



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO
EN INFORMÁTICA DE GESTIÓN

PHOTOUPLoad



DEPARTAMENTO: INGENIERÍA INFORMÁTICA

DIRECTOR DEL PROYECTO: MANUEL PALOMO DUARTE

AUTOR DEL PROYECTO: CARLOS ALBERTO BERNAL LÓPEZ

Cádiz, Noviembre 2013

Fdo.: Carlos Alberto Bernal López

Índice general

| | | |
|-------|---|----|
| 1 | Introducción..... | 10 |
| 1.1 | PhotoUpload | 10 |
| 1.2 | Estructura del documento | 11 |
| 2 | Conceptos básicos | 13 |
| 2.1 | ¿Qué es Wikipedia? | 13 |
| 2.2 | ¿Qué es Wikimedia Commons?..... | 13 |
| 2.3 | ¿Qué es Wiki Loves Monuments? | 14 |
| 2.4 | ¿Cómo utiliza la aplicación Wikipedia? | 14 |
| 2.5 | ¿Cómo utiliza la aplicación Wikimedia Commons?..... | 15 |
| 2.6 | ¿Cómo utiliza la aplicación Wiki Loves Monuments? | 15 |
| 2.7 | ¿Por qué esta aplicación? | 16 |
| 3 | Planificación | 17 |
| 3.1 | Incrementos:..... | 17 |
| 3.1.1 | Incremento 1 | 17 |
| 3.1.2 | Incremento 2 | 18 |
| 3.1.3 | Incremento 3 | 19 |
| 3.1.4 | Incremento 4 | 19 |
| 3.1.5 | Incremento 5 | 20 |
| 3.1.6 | Incremento 6 | 20 |
| 3.2 | Estadísticas..... | 21 |
| 3.2.1 | Tablas de tiempos | 21 |
| 3.2.2 | Diagramas de Gantt | 23 |
| 3.2.3 | Diagrama de sectores..... | 26 |
| 4 | Análisis | 30 |

Índice general

| | | |
|-------|---|----|
| 4.1 | Funcionalidades del sistema | 30 |
| 4.1.1 | La aplicación..... | 30 |
| 4.1.2 | Menú principal..... | 30 |
| 4.1.3 | Interfaces | 31 |
| 4.2 | Caso de uso | 31 |
| 4.2.1 | Menú principal..... | 32 |
| 4.2.2 | Interfaces | 34 |
| 4.2.3 | Ver información..... | 39 |
| 4.3 | Modelo conceptual de datos | 42 |
| 4.4 | Modelo de comportamiento | 46 |
| 4.4.1 | Menú principal..... | 46 |
| 4.4.2 | Ver info..... | 48 |
| 4.4.3 | Interfaces | 50 |
| 5 | Diseño..... | 57 |
| 5.1 | Diseño de la aplicación | 57 |
| 5.1.1 | Descripción de las clases | 57 |
| 6 | Implementación | 89 |
| 6.1 | API WIKIPEDIA | 89 |
| 6.2 | Multilenguaje | 90 |
| 6.3 | Obtener coordenadas..... | 90 |
| 6.4 | Leer el contenido de la API y tratar los datos..... | 90 |
| 6.5 | Mapa con monumentos | 91 |
| 7 | Pruebas | 92 |
| 7.1 | Incremento 1 | 92 |
| 7.2 | Incremento 2 | 93 |

Índice general

| | | |
|-------|---|-----|
| 7.3 | Incremento 3 | 93 |
| 7.4 | Incremento 4 | 94 |
| 7.5 | Incremento 5 | 95 |
| 7.6 | Prueba Completa..... | 96 |
| 8 | Conclusiones..... | 98 |
| 8.1 | Desarrollo del proyecto..... | 98 |
| 8.2 | Ampliaciones | 98 |
| 8.3 | Comparativa con Wikimedia Commons..... | 99 |
| 9 | Apéndice..... | 101 |
| 9.1 | Apéndice A | 101 |
| 9.1.1 | Instalación..... | 101 |
| 9.2 | Apéndice B | 102 |
| 9.2.1 | Guía de usuario..... | 102 |
| 10 | Bibliografía..... | 114 |
| | [5] Curso para Andorid. Todo lo que necesitas para empezar Autor: Maestro de la web | 114 |
| 11 | Agradecimientos..... | 117 |
| 12 | GNU Free Documentation License | 118 |

Índice de Figuras

| | |
|--|-----------|
| <i>Figura 1.1 Imagen portada aplicación.....</i> | <i>11</i> |
| <i>Figura 3.1 Diagrama de Gantt incremento 1</i> | <i>23</i> |
| <i>Figura 3.2 Diagrama de Gantt incremento 2</i> | <i>24</i> |
| <i>Figura 3.3 Diagrama de Gantt incremento 3</i> | <i>24</i> |
| <i>Figura 3.4 Diagrama de Gantt incremento 4.....</i> | <i>25</i> |
| <i>Figura 3.5 Diagrama de Gantt incremento 5</i> | <i>25</i> |
| <i>Figura 3.6 Diagrama de Gantt total incrementos</i> | <i>26</i> |
| <i>Figura 3.7 Diagrama de sectores incremento 1</i> | <i>26</i> |
| <i>Figura 3.8 Diagrama de sectores incremento 2</i> | <i>27</i> |
| <i>Figura 3.9 Diagrama de sectores incremento 3</i> | <i>27</i> |
| <i>Figura 3.10 Diagrama de sectores incremento 4.....</i> | <i>28</i> |
| <i>Figura 3.11 Diagrama de sectores incremento 5.....</i> | <i>28</i> |
| <i>Figura 3.12 Diagrama de sectores total de incrementos</i> | <i>29</i> |
| <i>Figura 4.1 Diagrama de caso de uso: Menú Principal</i> | <i>32</i> |
| <i>Figura 4.2 Diagrama de caso de uso: Interfaz</i> | <i>34</i> |
| <i>Figura 4.3 Diagrama de caso de uso: Ver información</i> | <i>39</i> |
| <i>Figura 4.4 Diagrama Conceptual de Datos</i> | <i>45</i> |
| <i>Figura 4.5 Diagrama de secuencia de sistema:Menu Principal</i> | <i>46</i> |
| <i>Figura 4.6 Diagrama de secuencia de sistema:ver información</i> | <i>48</i> |
| <i>Figura 4.7 Diagrama de secuencia de sistema:escoger opción</i> | <i>50</i> |
| <i>Figura 4.8 Diagrama de secuencia de sistema:rellenar formulario</i> | <i>52</i> |
| <i>Figura 4.9 Diagrama de secuencia de sistema:obtener datos</i> | <i>54</i> |
| <i>Figura 5.1 AndromediaCommonsActivity.....</i> | <i>57</i> |
| <i>Figura 5.2 Creación de ráfaga satisfactoria</i> | <i>59</i> |

Índice de figuras

| | |
|---|----|
| Figura 5.3 <i>Datos imagen wiki love</i> | 60 |
| Figura 5.4 <i>Datos imagen</i> | 61 |
| Figura 5.5 <i>Datos</i> | 63 |
| Figura 5.6 <i>Descripción licencia</i> | 65 |
| Figura 5.7 <i>Errores API</i> | 66 |
| Figura 5.8 <i>Licencias</i> | 67 |
| Figura 5.9 <i>Llamar URL</i> | 68 |
| Figura 5.10 <i>Llamar url</i> | 69 |
| Figura 5.11 <i>Login</i> | 69 |
| Figura 5.12 <i>Mapa monumentos</i> | 71 |
| Figura 5.13 <i>Mensaje campos obligatorios</i> | 72 |
| Figura 5.14 <i>Mensaje información</i> | 73 |
| Figura 5.15 <i>Menú información</i> | 74 |
| Figura 5.16 <i>Menú love monuments</i> | 75 |
| Figura 5.17 <i>Método subida</i> | 76 |
| Figura 5.18 <i>Modo captura</i> | 77 |
| Figura 5.19 <i>Obtener coordenadas</i> | 78 |
| Figura 5.20 <i>Selección distancia monumentos</i> | 80 |
| Figura 5.21 <i>Selección imagen individual</i> | 81 |
| Figura 5.22 <i>Selección imágenes ráfaga</i> | 83 |
| Figura 5.23 <i>Selección monumento</i> | 84 |
| Figura 5.24 <i>Selección ráfaga</i> | 85 |
| Figura 5.25 <i>Subida satisfactoria</i> | 86 |
| Figura 5.26 <i>Subir</i> | 87 |
| Figura 5.27 <i>Tomar foto</i> | 88 |

Índice de tablas

| | | |
|---------|---|----|
| Tabla 1 | <i>Estadística del incremento 1</i> | 21 |
| Tabla 2 | <i>Estadística del incremento 2</i> | 21 |
| Tabla 3 | <i>Estadística del incremento 3</i> | 22 |
| Tabla 4 | <i>Estadística del incremento 4</i> | 22 |
| Tabla 5 | <i>Estadística del incremento 5</i> | 22 |
| Tabla 6 | <i>Estadística total de incrementos</i> | 23 |

Índice de imágenes de la guía de usuario

| | |
|----------------------|-----|
| Imagen guía 1 | 103 |
| Imagen guía 2 | 103 |
| Imagen guía 3 | 104 |
| Imagen guía 4 | 105 |
| Imagen guía 5 | 106 |
| Imagen guía 6 | 106 |
| Imagen guía 7 | 107 |
| Imagen guía 8 | 107 |
| Imagen guía 9 | 108 |
| Imagen guía 10 | 108 |
| Imagen guía 11 | 109 |
| Imagen guía 12 | 109 |
| Imagen guía 13 | 110 |
| Imagen guía 14 | 111 |
| Imagen guía 15 | 111 |
| Imagen guía 16 | 112 |
| Imagen guía 17 | 113 |
| Imagen guía 18 | 113 |

Capítulo 1

1 Introducción

En este capítulo se hace un breve resumen de que es la aplicación que he desarrollado, las partes que tiene y sus funcionalidades. Y además detalla la estructura de este documento con una muy breve noción sobre que trata cada capítulo.

1.1 PhotoUpload

La telefonía móvil hace unos años que es prácticamente indispensable en nuestras vidas, y a lo largo de esta última década se han sucedido generaciones de móviles unas tras otras, desde los primeros móviles enormes que solo permitían llamar, pasando por los de pantalla a color o los que incluían cámara de fotos hasta los terminales que hay en nuestros días, los cuales nos permiten una gran variedad de posibilidades, tanto de conexiones, sistemas operativos y aplicaciones; que será el tema que nos abarque.

El objetivo de este PFC es desarrollar una aplicación para Android llamada PhotoUpload.

Esta aplicación consta de dos grandes bloques, el primero que se puede denominar subida de imágenes a WikiMedia Commons y el segundo se puede denominar actualización de monumentos de bien de interés cultural(BIC).

El primer bloque consta de varias funcionalidades como son la selección de imagen individual o imágenes en ráfaga, el posicionamiento mediante GPS o triangulación y la subida al repositorio de Wikimedia Commons del usuario.

El segundo bloque consta de varias funcionalidades como son la búsqueda de monumentos catalogados como bienes de interés cultural (BIC) sin imagen en un radio alrededor de usuario, la selección de este, su posterior subida al repositorio de WikiMedia Commons del usuario y la actualización del monumento en su listado de BIC con su imagen correspondiente.

Además durante el mes de Septiembre se habilita una funcionalidad adicional que permite participar en el concurso Wiki Loves Monuments.

A continuación se muestra e icono de la aplicación:



Figura 1.1 Imagen portada aplicación

1.2 Estructura del documento

Este documento se divide en 8 capítulos:

Capítulo 1: Introducción. – Capitulo actual.

Capitulo2: Conceptos básicos.- Se comentarán los aspectos esenciales tales como que recursos de Wikipedia utiliza éste programa y una descripción de ellos. Información sobre los servicios externos que utiliza la aplicación (Wikipedia, Wikimedia Commons y Wiki Loves Monuments) y la razón de por qué elegir mi aplicación frente a la de Wikipedia.

Capítulo 3: Planificación.-Se indican los pasos realizados en las distintas etapas por las que ha pasado el PFC y el tiempo que he invertido en cada una de ellas.

Capítulo 4: Análisis.- Se desarrollara el análisis de la aplicación y se propondrán distintos casos de usos para la aplicación.

Capítulo 5: Diseño.- Se llevara a cabo un estudio sobre el diseño de la aplicación y sus componentes.

Capítulo 6: Implementación.- Se trataran los aspectos más relevantes de la implementación de la aplicación, los cuales son la api de Wikipedia, el posicionamiento

Capítulo 1 Introducción

mediante GPS, tratar la devolución de datos de la api y pintar los monumentos en un mapa de google maps.

Capítulo 7: Problemas, soluciones y pruebas .- Se trataran las distintas pruebas que se realizan en cada incremento de la aplicación durante su desarrollo.

Capítulo 8: Conclusiones.- Expondré mis conclusiones y vivencias durante el periodo que he estado realizando el PFC.

Capítulo 2

2 Conceptos básicos

En este capítulo se detallan brevemente las aplicaciones en las que está apoyada mi aplicación y que uso hace de ellas. Se dice brevemente que es Wikipedia, Wikimedia Commons y Wiki Loves Monuments y que uso hace la aplicación de estos para ofrecer el servicio que presta.

2.1 ¿Qué es Wikipedia?

Según [1] “**Wikipedia** es una enciclopedia libre, políglota y editada colaborativamente. Es administrada por la Fundación Wikimedia, una organización sin ánimo de lucro. Sus más de 37 millones de artículos en 284 idiomas (cantidad que incluye dialectos de muchos de esos idiomas) han sido redactados conjuntamente por voluntarios de todo el mundo, y prácticamente cualquier persona con acceso al proyecto puede editarlos. Iniciada en enero de 2001 por Jimmy Wales y Larry Sanger, es la mayor y más popular obra de consulta en Internet.”

2.2 ¿Qué es Wikimedia Commons?

Según [2] “**Wikimedia Commons** (frecuentemente abreviado como **Commonswiki** o simplemente **Commons**) es un proyecto de la Fundación Wikimedia creado el 7 de septiembre de 2004, cuyo objetivo es servir como almacén de archivos de imágenes y otros formatos multimedia para el resto de los proyectos de la fundación. Todos los archivos deben haber sido liberados bajo licencias libres (como la licencia de documentación libre de GNU) o en el dominio público, actualmente tiene más de trece millones de archivos multimedia libres.

Se basa en un wiki configurado de tal forma que los archivos allí contenidos están disponibles en todos los demás wikis de los proyectos de Wikimedia de forma transparente. Esto permite una gestión centralizada de este tipo de archivos en todas las versiones de Wikipedia, Wikilibros, etc.”

2.3 ¿Qué es Wiki Loves Monuments?

Según [3] “**Wiki Loves Monuments** es un **concurso fotográfico de monumentos** abierto a todo el mundo. Está organizado por los capítulos Wikimedia y grupos de voluntarios en distintos países. En España está organizado por la asociación Wikimedia España. La Fundación Wikimedia es la entidad que da soporte a Wikipedia, la enciclopedia libre, entre otros proyectos.

El Patrimonio Cultural es una parte importante del conocimiento que recoge y difunde Wikipedia. Todo el mundo puede contribuir tanto con imágenes como con artículos. Una imagen vale más que mil palabras, en todos los idiomas.

Todo el mundo está invitado a cargar imágenes de monumentos bajo una licencia libre para su uso en las Wikipedias. La idea de este proyecto es conseguir una visión global y única del patrimonio cultural de cada país.”

2.4 ¿Cómo utiliza la aplicación Wikipedia?

Wikipedia alberga un listado de bienes de interés cultural de inmuebles y esta aplicación recoge los monumentos catalogados como bienes de interés cultural que no poseen una imagen en dicho listado para que el usuario que esté cerca de este monumento puede proporcionarla si lo desea.

De manera más concreta, la interacción entre aplicación (PhotoUpload) y plataforma (Wikipedia) es la siguiente:

El usuario busca monumentos en un radio alrededor de él, la aplicación mediante la API de Wikipedia solicita un listado de momentos sin imágenes en el radio seleccionado. La API devuelve una serie de monumentos que la aplicación se encarga de estructurar y mostrar para que el usuario pueda seleccionar el que desee insertar su imagen. Y finalmente cuando la imagen es subida a WikiMedia Commons, la aplicación se encarga de actualizar el listado de BICs mediante la API de Wikipedia, mandándole a esta el mismo listado que la API le devolvió anteriormente pero con la referencia a la nueva imagen incluida.

2.5 ¿Cómo utiliza la aplicación Wikimedia Commons?

Wikimedia Commons es el repositorio donde se guardan las imágenes que el usuario ha tomado y subido de manera correcta con esta aplicación, asignándole un título, una descripción y una licencia entre otras características. A partir de este momento las imágenes pasaran a ser de dominio público.

De manera más concreta, la manera de interactuar entre la aplicación y WikiMedia Commons es la siguiente:

El usuario toma una imagen de su galería y tras introducir todos los datos pertinentes, envía a WikiMedia Commons mediante una consulta a su API los datos de la imagen que hemos introducido en la aplicación, así como el nombre de esta, su ruta y el archivo lógico en sí.

2.6 ¿Cómo utiliza la aplicación Wiki Loves Monuments?

Durante el mes de Septiembre la aplicación activa una funcionalidad adicional que permite a los usuarios participar en el concurso sin necesidad de tener que ir directamente a la web a subir la imagen y ofreciendo la oportunidad de enriquecer aún más tanto su propio repositorio como el contenido global de archivos multimedia.

De manera más concreta, la manera de interactuar la aplicación con Wiki Loves Monuments es la siguiente:

Cuando tenemos la imagen seleccionada y decimos que la vamos a subir para participar en Wiki Loves Monuments, la aplicación añade en las propiedades de la imagen una etiqueta definida por Wiki Loves Monuments que indica que dicha imagen participará en el concurso para el año actual.

2.7 ¿Por qué esta aplicación?

Hasta hace unos meses no existía una aplicación poder subir a dicho repositorio sus imágenes de forma inmediata, lo habitual era que hicieses tus fotos con el móvil o cámara convencional y luego una vez estas en casa te tenías que sentar delante del ordenador y añadirlas al repositorio de una en una, y además con la dificultad de obtener las coordenadas del lugar donde habías tomado la imagen anteriormente en caso de que la quisieras tener localizada en el mapa.

Por otro lado, para actualizar una imagen del listado de Bienes de Interés Cultural BIC, debías de buscar dicho listado, he ir al monumento en concreto y añadir una imagen.

Ahora con esta aplicación todo es más sencillo, puedes subir a tú repositorio de Wikimedia Commons de manera inmediata desde el móvil en el momento que tomas la foto y además posicionarla de manera automática para tenerla localizada en el mapa.

Además incluye la posibilidad de si tienes varias imágenes de una temática concreta o de cualquier otra índole, puedes crear una ráfaga de imágenes y subirla todas de manera automática en solo tres pasos.

El por qué elegir mi aplicación y no la de Wikimedia Commons, la resumo en la versatilidad que está permite, mientras que la aplicación de Wikimedia Commons solo permite subir imágenes al repositorio de una en una, mi aplicación permite subir tanto de una en una como crear una ráfaga y luego subirla de forma automática. Esto da una ventaja a mi aplicación, y es que agiliza el tiempo que dedicarás a dicho proceso.

Además la aplicación de Wikimedia Commons solo te permite subir y visualizar dichas imágenes, mientras que mi aplicación te añade las coordenadas de tú posición. Además te permite visionar un mapa con tus imágenes subidas al repositorio y sus datos.

Y para decantarme aún más por mi aplicación, además de todo lo anteriormente mencionado, la aplicación da la oportunidad de contribuir a Wikipedia colaborando de manera desinteresada en la actualización de los listado de bienes de interés cultura, ya que si haces una búsqueda de estos, desde donde te encuentres, puedes tener la oportunidad de actualizar algún monumento que carezca de imagen.

Y por último, mi aplicación da la oportunidad de en el mes de Septiembre participar en el concurso Wiki Loves Monuments.

Capítulo 3

3 Planificación

En este capítulo se detalla la planificación de cada una de las etapas en las que dividí el desarrollo de la aplicación, los problemas más importantes con los que me encontré y el tiempo aproximado que he dedicado a cada funcionalidad. También acompaño estos datos con una serie de estadísticas en tablas de tiempo, diagramas de Gantt y de sectores para reflejar en que he invertido el tiempo en cada incremento.

3.1 Incrementos:

Decidí utilizar un modelo incremental debido a que nunca había realizado nada de esta envergadura y por lo tanto decidí ir haciendo funcionalidades de la aplicación por separado hasta que estuviesen operativas y luego unir las para obtener el proyecto final.

Tras establecer unos criterios que me marcaran cuales eran los pasos a seguir (hasta donde llegar con cada incremento) para obtener la última versión de la aplicación decidí hacer 5 etapas o incrementos.

3.1.1 Incremento 1

Me planteé como debería de ser el inicio de la aplicación ya que debía de estar logado con un usuario y un password para poder subir imágenes y Wikimedia Commons.

Al principio pensé en crearme una clase con unos métodos que fuesen los encargados de hacer login y logout. Por lo tanto estudié de qué manera se hacía login con la documentación y las diversas aplicaciones que me ofrecía Wikipedia. Tras muchas horas intentándolo, aproximadamente (120 horas) conseguí que me hiciese login, aunque no funcionaba bien ya que a veces si se hacía y otras no, investigando que era lo que me sucedía encontré un código en java con todos los métodos necesarios para comunicarme con la API de Wikipedia por lo que lo único que tenía que hacer era cambiar el dominio y ya hacía login. Después de esto, ya pude adaptar la clase que encontré con sus métodos y variables a mi problema para hacer más fácil el acceso a lo que yo necesitaba.

Capítulo 3 Planificación

Esta clase que trata con la API de Wikipedia tiene 69 atributos, 3 constructores y 161 métodos, muchos de los cuales no serán utilizados y por lo tanto se descartarán en función de que vayan a pertenecer o no a la aplicación.

Por otro lado, cree otra clase para mantener los datos necesarios para la subida de la imagen en memoria principal, sin necesidad de crear ficheros que mantuviesen los datos o una base de datos para este fin. Esto me llevo aproximadamente unas 15 horas.

También en este primer incremento cree las interfaces para la subida de imágenes tanto manual como automáticamente. Estas interfaces son un total 8 con una estructura muy sencilla que consta de botones para navegar por el menú, cuadros de textos para la introducción de datos y visualización de texto. La creación de las distintas interfaces me llevo aproximadamente unas 20 horas.

Asociada a estas interfaces va el código que crea la lógica deseada para las interfaces y la cual tardé en desarrollar unas 50 horas aproximadamente.

Por lo tanto tarde aproximadamente 205 horas en tener una primera versión operativa de la aplicación dedicando una media diaria de 3,25 horas.

Las funcionalidades disponibles en este incremento eran las de loguearse, seleccionar una imagen de la galería, escoger el modo de subida entre semiautomático o manual, la elección de la licencia que se le otorga a la imagen y detallar su descripción para que sirviese a modo ilustrativo para el usuario, rellenar el formulario de datos relacionados con la imagen (título, descripción corta, descripción, categoría y fecha) y por ultimo subir la imagen a WikiMedia Commons.

3.1.2 Incremento 2

En este incremento realicé las funcionalidades de geolocalizar las imágenes mediante la señal GPS captada por el móvil o mediante triangulación cuando está conectado a internet, además creé la interfaz para utilizar dicha funcionalidades y asignar a las imágenes tomadas del incremento anterior los datos actuales.

Capítulo 3 Planificación

También creé en este incremento la interfaz que te muestra por pantalla los posibles errores que pueda haber en el transcurso de la aplicación y su lógica.

Además realice las modificaciones necesarias para poder integrar ambos incremento y obtener otra versión operativa más compleja.

Este incremento duró aproximadamente del 16 de Abril al 15 de Mayo de 2012.

3.1.3 Incremento 3

En este incremento realicé la lógica de crear una ráfaga de imágenes y también subirlas de forma automática.

Además cree las interfaces necesarias para llevar a cabo la lógica antes comentada y las modificaciones necesarias en la aplicación resultante del segundo incremento para obtener otra nueva versión final.

Este incremento tuvo una duración aproximada de 1 mes entre el 16 de Mayo y el 15 de Junio de 2012 aproximadamente y dedicando cada día unas 3 horas de lunes a viernes.

3.1.4 Incremento 4

En este incremento realice el desarrollo para buscar monumentos tanto en anexos como en listas de Wikipedia que no tuvieran una imagen asociada.

Los pasos que he seguí para llegar hasta aquí fueron los siguientes:

Cree una clase que me recogiese los datos de la API de Wiki Loves Monuments asociados a los monumentos sin imágenes a partir de una distancia determinada alrededor de la ubicación del dispositivo móvil.

Por otro lado, cree una clase que en la que desarrolle las funcionalidades necesarias para mostrar por pantalla los nombre y distancias de los monumentos que se encuentran en la distancia anteriormente seleccionadas.

Y por último cree otra clase que te abre la cámara y permite hacer la foto al monumento correspondiente. Tras esto, automáticamente la imagen se sube a su listado o anexo correspondiente.

Capítulo 3 Planificación

He tardado aproximadamente un mes en realizar este incremento, fue durante Julio de 2012 y dediqué 3 horas diarias de lunes a viernes.

3.1.5 Incremento 5

En este incremento desarrolle una clase que en función de los datos de los monumentos obtenidos en un radio determinado, cogiese sus coordenadas y me las pintase en un mapa con una imagen y que se pudiese hacer click en ella para ver que monumento es y a que distancia se encuentra.

En este desarrollo tarde 10 días dedicándole 3 horas por cada uno de ellos desde el 20 de Agosto, hasta el 30 de este mismo mes.

3.1.6 Incremento 6

En el último incremento creé dos funcionalidades, la primera es la inserción en una base de datos las imágenes subidas con la aplicación y la segunda consiste en leer la base de datos, y mostrar un mapa con las imágenes que el usuario ha subido a Wikimedia Commons.

La base de datos que utilizo es muy sencilla, ya que solo contiene una tabla en la cual guardo los datos de las imágenes que subo con la aplicación a Wikimedia Commons. La manera en la que utiliza la aplicación la base de datos es muy sencilla, cada vez que se sube una imagen, antes de volver al menú principal, se hace un insert en la tabla con los datos de dicha imagen.

Por otro lado, la funcionalidad de mostrar los monumentos subidos a Wikimedia Commons con la aplicación consiste en leer de la base de datos los monumentos insertados y pintarlos en un mapa. Además incluye la opción de pulsar sobre ellos y que te aparezca la imagen que subiste junto a los datos que introdujiste para su subida.

Además y para enriquecer un poco más la aplicación se desarrollaron dos nuevas funcionalidades relacionadas con los datos de la imagen, se implementó un método que ahora permite introducir más de una categoría a la imagen que se subirá a WikiMedia

Capítulo 3 Planificación

Commons. Y por último se desarrolló otro método que evalúa la fecha en la que se tomó la imagen y la pone como fecha por defecto para la subida. Anteriormente por defecto tomaba la fecha del día actual.

3.2 Estadísticas

A continuación muestro las tablas con las estadísticas de cada uno de los incrementos, sus diagramas de Gantt correspondiente y los diagramas de sectores con el total de horas invertidas en cada actividad del incremento.

3.2.1 Tablas de tiempos

| Actividades | Fecha Inicio | Horas Invertidas | Fecha Fin | Días |
|---------------------------|--------------|------------------|------------|------|
| Formación | 15/02/2012 | 30 | 25/02/2012 | 10 |
| Buscar Información | 20/02/2012 | 20 | 05/03/2012 | 15 |
| Programación e interfaces | 01/03/2012 | 145 | 12/04/2012 | 43 |
| Pruebas | 12/04/2012 | 10 | 15/04/2012 | 4 |

Tabla 1 Estadística del incremento 1

| Actividades | Fecha Inicio | Horas invertidas | Fecha Fin | Días |
|---------------------------|--------------|------------------|------------|------|
| Formación | 16/04/2012 | 0 | 16/04/2012 | 0 |
| Buscar Información | 16/04/2012 | 10 | 20/04/2012 | 5 |
| Programación e interfaces | 18/04/2012 | 65 | 12/05/2012 | 27 |
| Pruebas | 12/05/2012 | 3 | 15/05/2012 | 4 |

Tabla 2 Estadística del incremento 2

| Actividades | Fecha Inicio | Horas Invertidas | Fecha Fin | Días |
|---------------------------|--------------|------------------|------------|------|
| Formación | 16/05/2012 | 0 | 16/05/2012 | 0 |
| Buscar Información | 16/05/2012 | 0 | 16/05/2012 | 0 |
| Programación e interfaces | 16/05/2012 | 43 | 15/06/2012 | 31 |
| Pruebas | 15/06/2012 | 2 | 16/06/2012 | 2 |

Tabla 3 Estadística del incremento 3

| Actividades | Fecha Inicio | Horas Invertidas | Fecha Fin | Días |
|---------------------------|--------------|------------------|------------|------|
| Formación | 01/07/2012 | 0 | 01/07/2012 | 0 |
| Buscar Información | 01/07/2012 | 3 | 03/07/2012 | 3 |
| Programación e interfaces | 01/07/2012 | 60 | 28/07/2012 | 28 |
| Pruebas | 20/07/2012 | 37 | 28/07/2012 | 9 |

Tabla 4 Estadística del incremento 4

| Actividades | Fecha Inicio | Horas Invertidas | Fecha Fin | Días |
|---------------------------|--------------|------------------|------------|------|
| Formación | 22/08/2012 | 0 | 22/08/2012 | 0 |
| Buscar Información | 22/08/2012 | 10 | 25/08/2012 | 4 |
| Programación e interfaces | 23/08/2012 | 50 | 20/09/2012 | 32 |
| Pruebas | 18/09/2012 | 10 | 21/09/2012 | 4 |
| Documentación | 22/09/2012 | 55 | 12/11/2012 | 52 |

Tabla 5 Estadística del incremento 5

| Incrementos | Fecha Inicio | Horas Invertidas | Fecha Fin | Días |
|--------------|--------------|------------------|------------|------|
| Incremento 1 | 15/02/2012 | 205 | 15/04/2012 | 72 |
| Incremento 2 | 16/04/2012 | 78 | 15/05/2012 | 36 |
| Incremento 3 | 16/05/2012 | 45 | 16/06/2012 | 33 |
| Incremento 4 | 01/07/2012 | 100 | 28/07/2012 | 40 |
| Incremento 5 | 22/08/2012 | 125 | 12/11/2012 | 92 |

Tabla 6 *Estadística total de incrementos*

3.2.2 Diagramas de Gantt

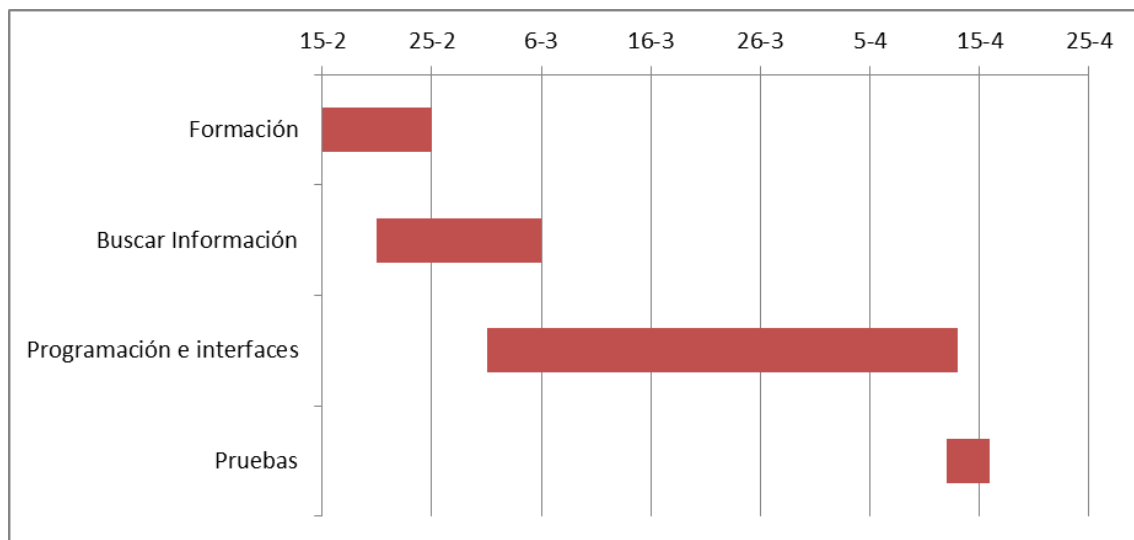


Figura 3.1 *Diagrama de Gantt incremento 1*

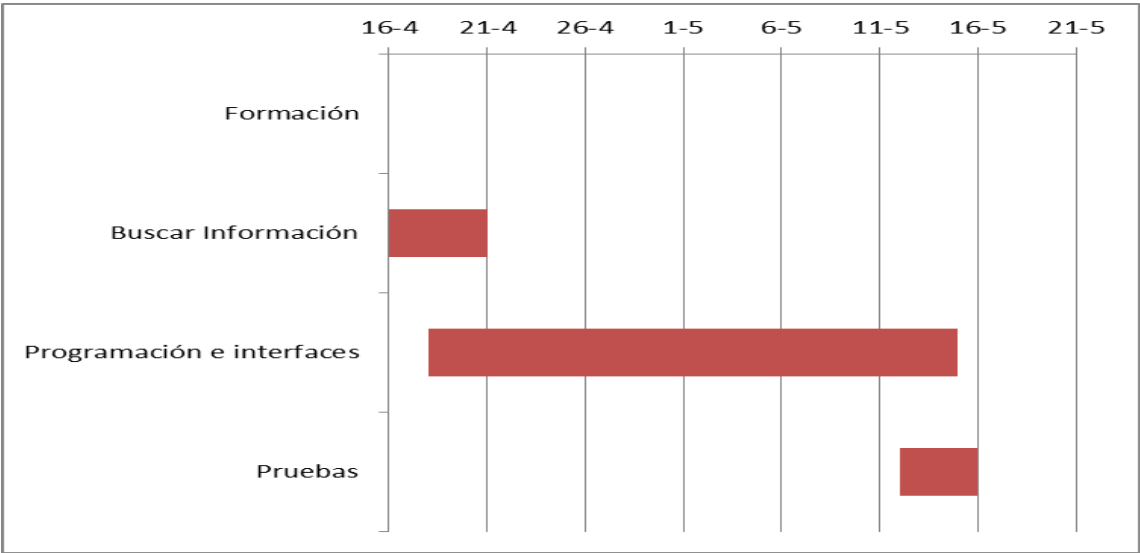


Figura 3.2 Diagrama de Gantt incremento 2

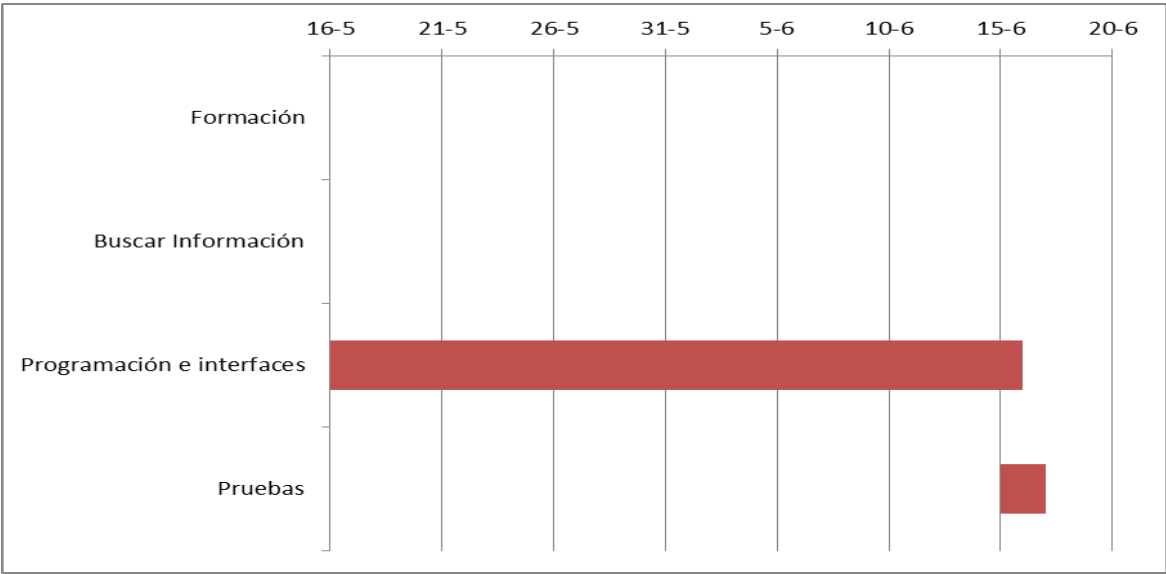


Figura 3.3 Diagrama de Gantt incremento 3

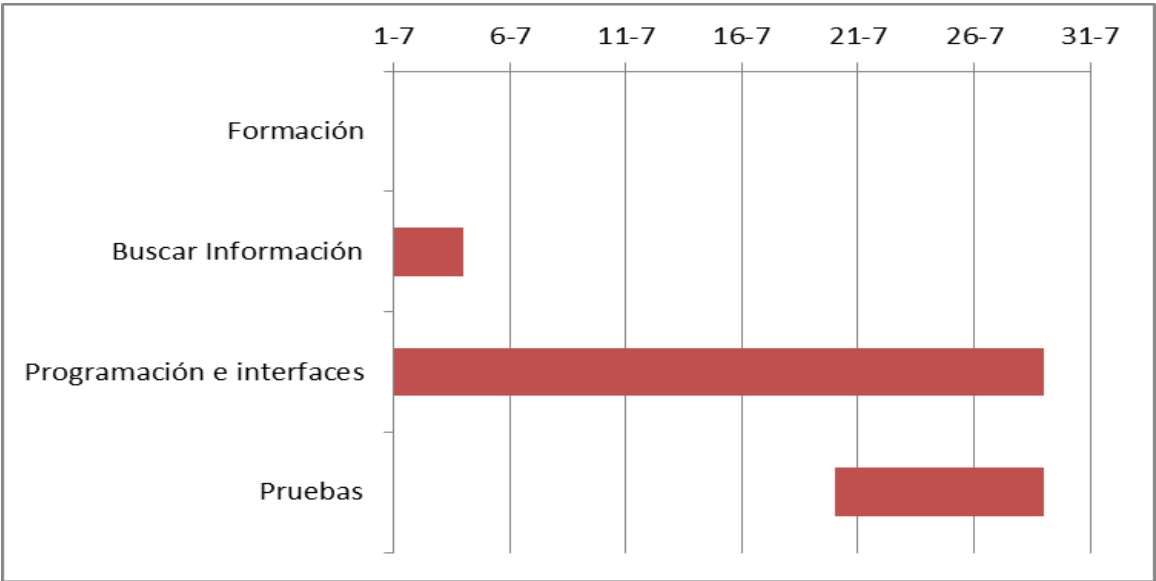


Figura 3.4 Diagrama de Gantt incremento 4

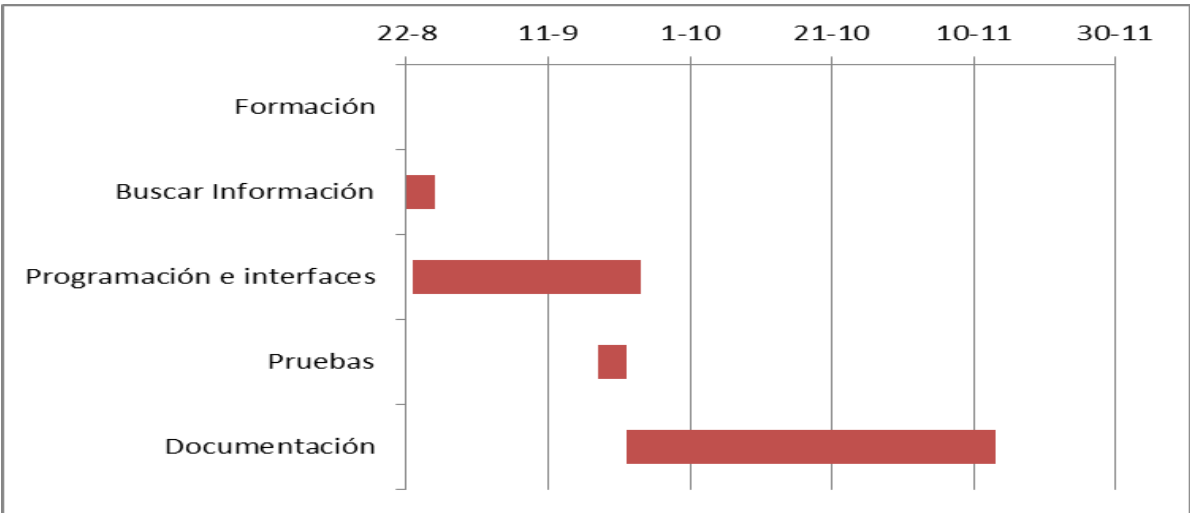


Figura 3.5 Diagrama de Gantt incremento 5

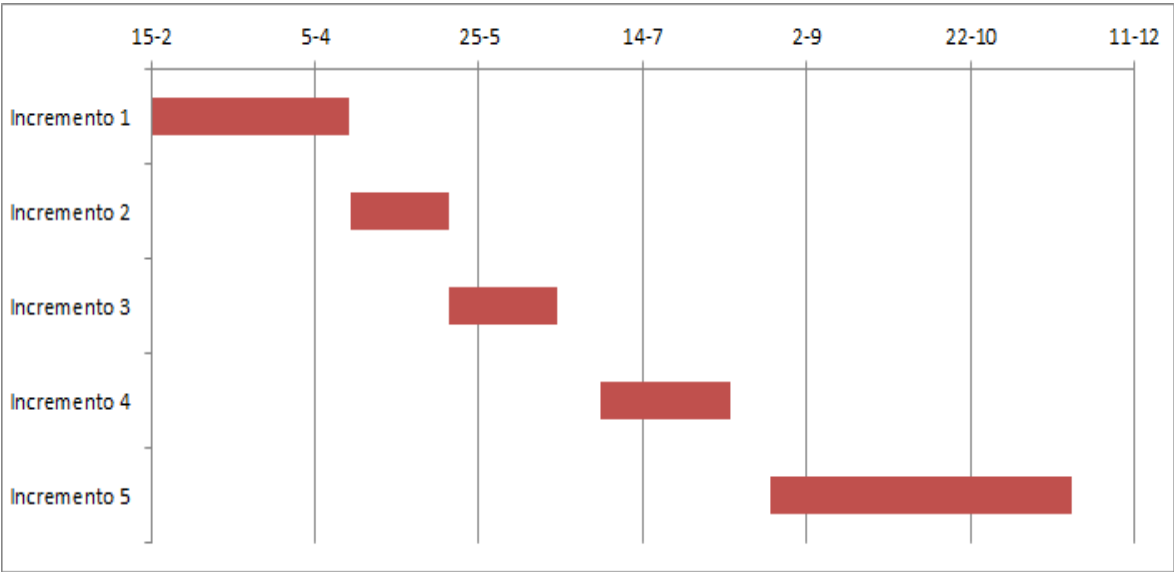


Figura 3.6 Diagrama de Gantt total incrementos

3.2.3 Diagrama de sectores

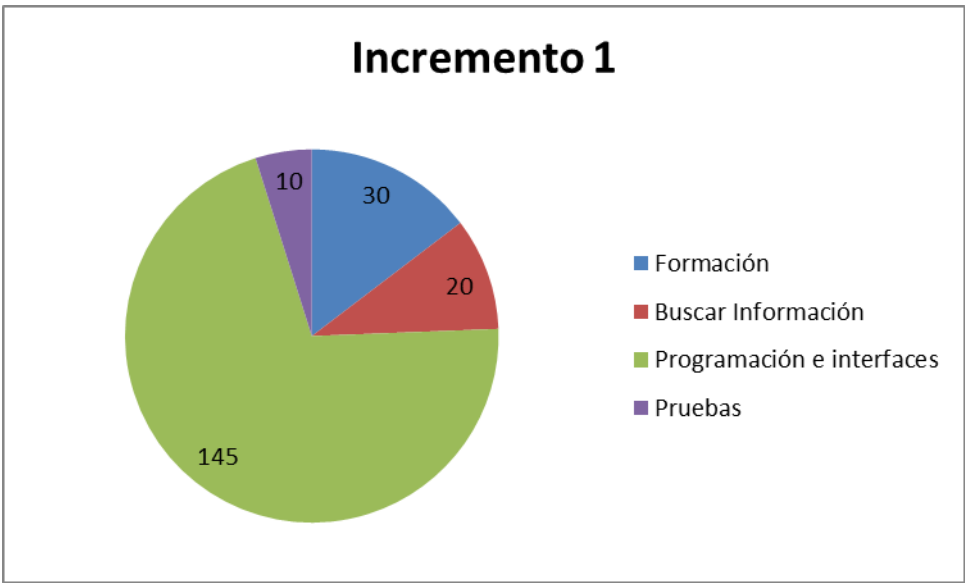


Figura 3.7 Diagrama de sectores incremento 1

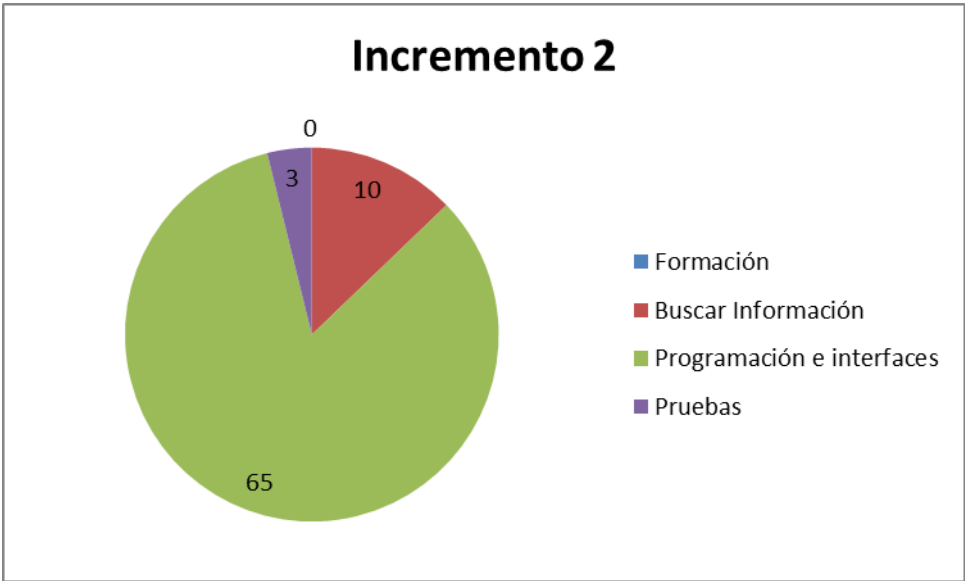


Figura 3.8 Diagrama de sectores incremento 2

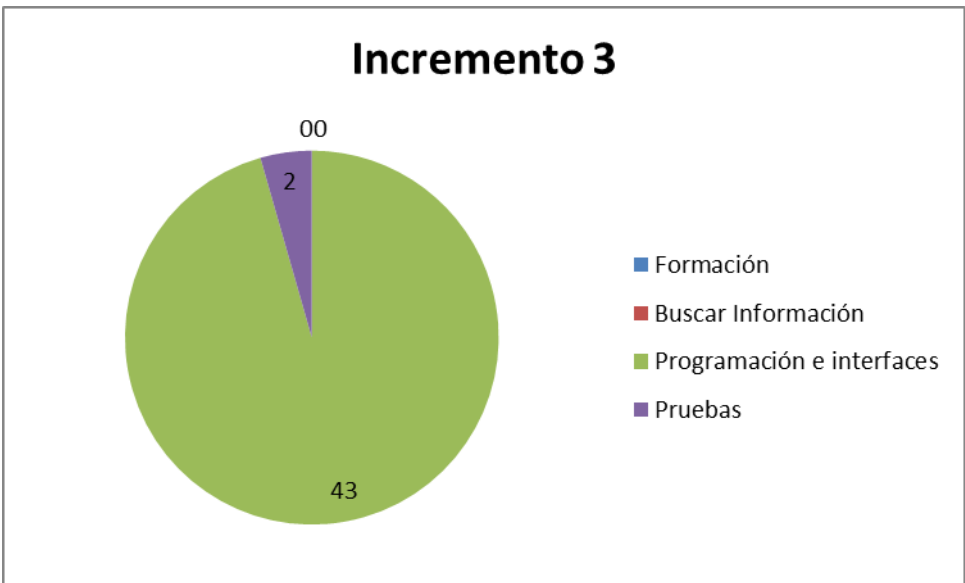


Figura 3.9 Diagrama de sectores incremento 3

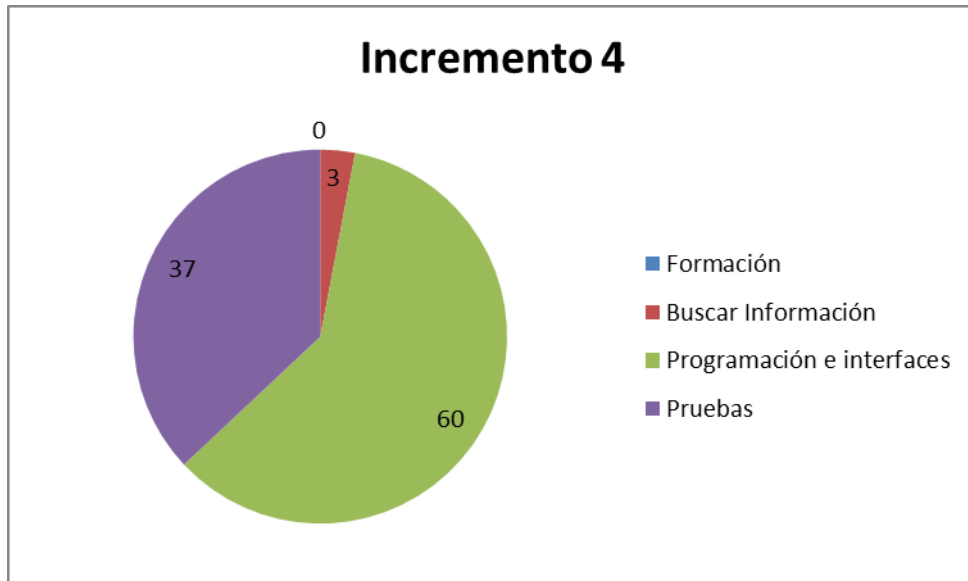


Figura 3.10 Diagrama de sectores incremento 4

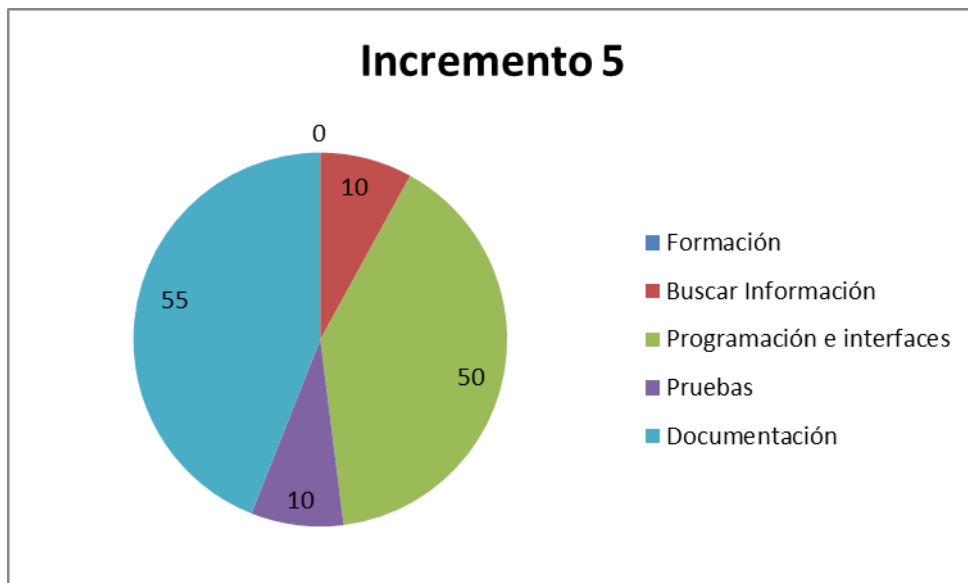


Figura 3.11 Diagrama de sectores incremento 5

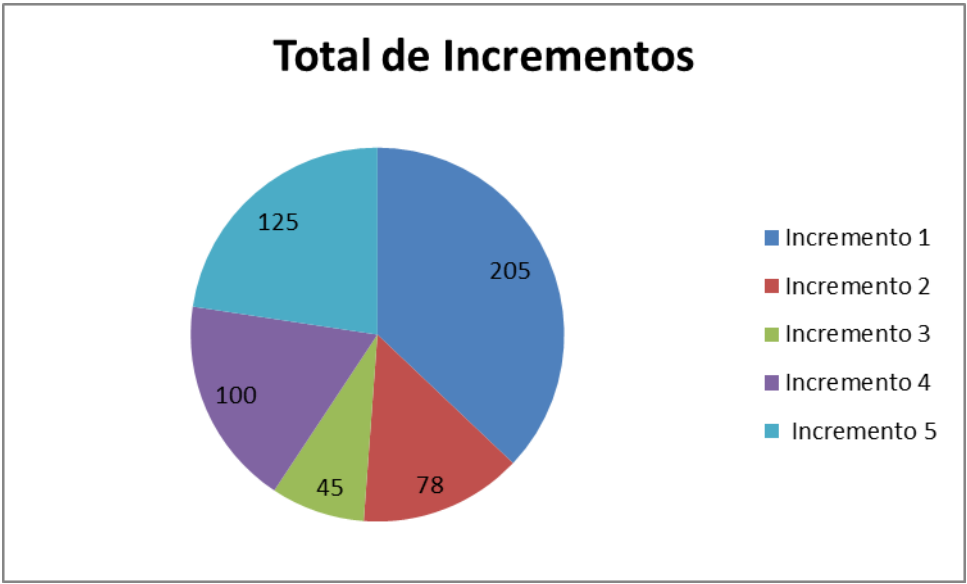


Figura 3.12 Diagrama de sectores total de incrementos

Capítulo 4

4 Análisis

En este capítulo se lleva a cabo una descripción al detalle de los componentes de las actividades que realiza la aplicación. También se detallan los casos de usos en los que se puede ver la interacción entre el usuario y la aplicación y las posibles salidas que el sistema muestra según el caso.

Además se desarrolla el modelo conceptual de datos en el cual se reflejan las estructuras de datos utilizadas y que funcionalidad realizan en la aplicación.

Y por último este tema incluye el modelo de comportamiento que especifica cómo debe de actuar la aplicación en cada uno de los casos en los que se puede encontrar.

4.1 Funcionalidades del sistema

A continuación se describen los elementos de la aplicación y su implementación.

4.1.1 *La aplicación*

La aplicación consta de varios objetos, cada uno de estos objetos tiene una interfaz y una o varias clases encargadas de la lógica de la aplicación.

Podemos hacer dos grandes bloques en los que englobar los objetos que forman parte de la aplicación, los primeros serían los que gestionarían el menú principal y los demás serán los que generan las interfaces.

4.1.2 *Menú principal*

El menú principal es cuando se inicia la aplicación, y podrás acceder al contenido de la aplicación, ver información de utilidad o salir de la aplicación (botón del teléfono).

Cuando la aplicación comience, se creará la clase `AndromediaCommons`, que es la encargada de crear la interfaz del menú principal, clase `API_Wiki` que se encarga de gestionar la comunicación con la API de Wikipedia.

Capítulo 4 Análisis

Además se crean las clases `datos`, `datos_imagen_wikilove` y `datos_imagen_rafaga` que son las encargadas de gestionar una serie de parámetros para mantener en memoria principal los datos necesarios durante la ejecución de la aplicación.

4.1.3 Interfaces

La aplicación está formada por varias interfaces, las cuales están definidas en ficheros XML y gestionadas por su clase correspondiente cada una como son: `licencias`, `método_subida`, `modo_captura`, etc...

Un usuario en un momento determinado solo puede estar en una única interfaz, de aquí que solo se trate una interfaz en cada comento.

4.2 Caso de uso

Ahora se especifican los casos de usos que describen el correcto funcionamiento de la aplicación.

He pensado 3 casos de usos los cuales abarcan todas las opciones de donde se puede encontrar la aplicación en cualquier momento: menú principal, Viendo la información o moverte por una de las interfaces.

4.2.1 Menú principal

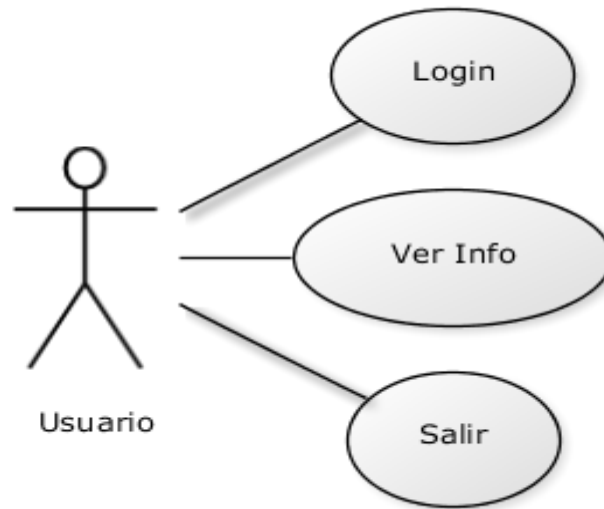


Figura 4.1 *Diagrama de caso de uso: Menú Principal*

Caso de uso: Login

Actor principal: Usuario.

Precondiciones: La aplicación está en el menú principal.

Postcondiciones: Se arranca la aplicación.

Escenario principal:

1. El usuario presiona el botón login
2. Se sale del menú principal.
3. Se muestra la siguiente interfaz “Elige modo de captura”

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.
- * El usuario no está logeado.
- * El usuario no tiene cuenta creada.

Capítulo 4 Análisis

- * El usuario no introduce las credenciales correctamente.
- * Error derivado de incompatibilidad entre la versión del sistema y la de la aplicación.
- * Error derivado de no tener conexión a internet.

Caso de uso: Ver Info.

Actor principal: Usuario.

Precondiciones: La aplicación está en el menú principal.

Postcondiciones: Se accede al menú de opciones de información.

Escenario principal:

1. El usuario presiona el botón Info.
2. Se sale del menú principal.
3. Se muestra el menú de información con sus distintas opciones.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.

Caso de uso: Salir.

Actor principal: Usuario.

Precondiciones: La aplicación está en el menú principal.

Postcondiciones: Se sale al me menú del teléfono cerrando la aplicación (gestionado por Android).

Escenario principal:

1. El usuario Presiona el botón HOME del teléfono o la flecha de atrás.
2. Se sale de la aplicación.

Escenario alternativos o de error: No existe.

4.2.2 Interfaces

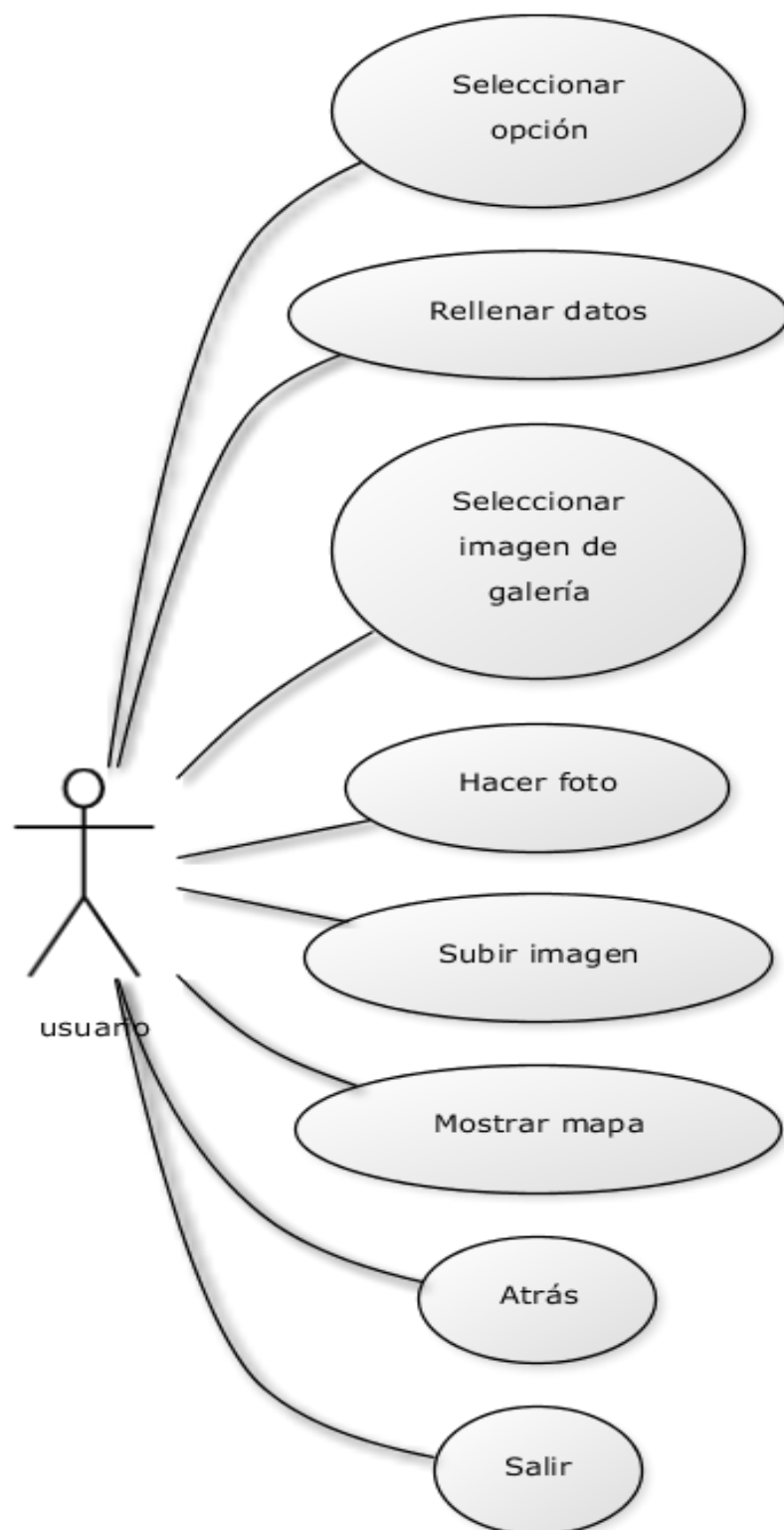


Figura 4.2 Diagrama de caso de uso: Interfaz

Capítulo 4 Análisis

Caso de uso: Seleccionar opción

Actor principal: Usuario.

Precondiciones:

- * La aplicación está en una interfaz determinada.
- * El usuario ha cumplimentado correctamente las interfaces anteriores.
- * El usuario ha cumplimentado los datos de la interfaz actual (si se requiere).
- * La interfaz actual no es la de subida satisfactoria.

Postcondiciones: Se pasa a la siguiente interfaz.

Escenario principal:

1. El usuario presiona el botón Siguiente u opción correspondiente.
2. Se sale de la interfaz actual.
3. Se muestra la siguiente interfaz que corresponde.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.
- * Faltan datos obligatorios por cumplimentar.
- *Error derivado del teléfono quedarse sin recursos disponibles para dar soporte a la aplicación.
- * Error derivado de no tener conexión a internet.

Caso de uso: Rellenar datos

Actor principal: Usuario.

Precondiciones:

- * La aplicación está en una interfaz determinada.
- * La interfaz actual no es el menú principal
- * La aplicación no está a la espera de recibir datos (escuchando)
- * La interfaz actual no es la de subida satisfactoria.

Capítulo 4 Análisis

Postcondiciones: Se pasa a la siguiente interfaz.

Escenario principal:

1. El usuario presiona rellena los datos que se requieren.
2. Se sale de la interfaz actual.
3. Se muestra la siguiente interfaz que corresponde.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.
- * El usuario no rellena los campos obligatorios
- *Error derivado del teléfono quedarse sin recursos disponibles para dar soporte a la aplicación.

Caso de uso: Seleccionar imagen de galería

Actor principal: Usuario.

Precondiciones:

- * La aplicación está en una interfaz determinada.
- * La interfaz actual no es el menú principal
- * La aplicación no está a la espera de recibir datos (escuchando)
- * La interfaz actual no es la de subida satisfactoria.
- * La galería de imágenes del teléfono no está vacía.

Postcondiciones: Se pasa a la siguiente interfaz con la imagen seleccionada.

Escenario principal:

1. El usuario presiona sobre la imagen que desea seleccionar.
2. Se sale de la interfaz actual.
3. Se muestra la siguiente interfaz que corresponde con la imagen seleccionada.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.

Capítulo 4 Análisis

- * El usuario no selecciona ninguna imagen.
- *Error derivado del teléfono quedarse sin recursos disponibles para dar soporte a la aplicación.

Caso de uso: Hacer foto.

Actor principal: Usuario.

Precondiciones:

- * La aplicación está en una interfaz determinada.
- * La interfaz actual no es el menú principal
- * La aplicación no está a la espera de recibir datos (escuchando)
- * La interfaz actual no es la de subida satisfactoria.
- * Hay memoria suficiente en el teléfono para almacenar una nueva imagen.

Postcondiciones: Se pasa a la siguiente interfaz con la imagen tomada.

Escenario principal:

1. El usuario presiona sobre el botón hacer foto de la cámara de la móvil.
2. Se sale de la interfaz actual.
3. Se muestra la siguiente interfaz que corresponde con la imagen tomada.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.
- * El usuario no hace ninguna foto.
- * Error derivado del teléfono quedarse sin recursos disponibles para dar soporte a la aplicación.
- * Memoria insuficiente en el teléfono.

Capítulo 4 Análisis

Caso de uso: Atrás

Actor principal: Usuario.

Precondiciones:

- * La aplicación está en una interfaz determinada.
- * La interfaz actual no es el menú principal
- * La aplicación no está a la espera de recibir datos (escuchando)
- * La interfaz actual no es la de subida satisfactoria.

Postcondiciones: Se pasa a la interfaz anterior.

Escenario principal:

1. El usuario presiona el botón Atrás.
2. Se sale de la interfaz actual.
3. Se muestra la anterior interfaz que corresponde.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.
- *Error derivado del teléfono quedarse sin recursos disponibles para dar soporte a la aplicación.
- * Error derivado de no tener conexión a internet.

Caso de uso: Salir

Descrito en el apartado 4.2.1

4.2.3 Ver información.

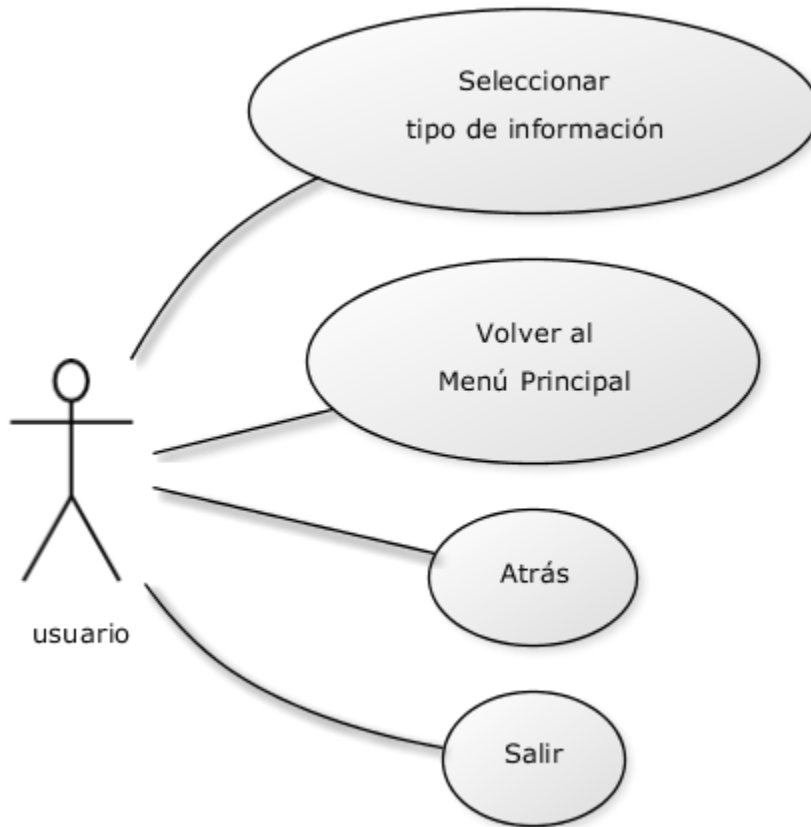


Figura 4.3 Diagrama de caso de uso: Ver información

Caso de uso: Seleccionar tipo de información

Actor principal: Usuario.

Precondiciones:

- * La aplicación está en una interfaz determinada.
- * La interfaz actual no es el menú principal
- * La aplicación no está a la espera de recibir datos (escuchando)
- * La interfaz actual no es la de subida satisfactoria.

Capítulo 4 Análisis

Postcondiciones: Se pasa a la interfaz seleccionada.

Escenario principal:

1. El usuario presiona el uno de los tipos de información ofrecidos.
2. Se sale de la interfaz actual.
3. Se muestra la anterior interfaz que corresponde.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.
- *Error derivado del teléfono quedarse sin recursos disponibles para dar soporte a la aplicación.

Caso de uso: Volver al Menú Principal

Actor principal: Usuario.

Precondiciones:

- * La aplicación está en una interfaz determinada.
- * La interfaz actual no es el menú principal
- * La aplicación no está a la espera de recibir datos (escuchando)
- * La interfaz actual no es la de subida satisfactoria.

Postcondiciones: Se pasa a la interfaz seleccionada.

Escenario principal:

1. El usuario presiona botón de Menú Principal
2. Se sale de la interfaz actual.
3. Se muestra la anterior interfaz que corresponde.

Escenarios alternativos o de error:

- * El usuario cierra la aplicación.
- *Error derivado del teléfono quedarse sin recursos disponibles para dar soporte a la aplicación.

Capítulo 4 Análisis

Caso de uso: Atrás

Descrito en apartado 4.2.2

Caso de uso: Salir

Descrito en apartado 4.2.1

4.3 Modelo conceptual de datos

Para realizar la aplicación he creado o reutilizado las siguientes clases:

- **AndromediaCommonsActivity:** Clase principal de la aplicación. Se encarga de crear la interfaz del menú principal, controlar su lógica y además crea las clases `API_Wiki` y `datos_imagen_wiki_love`.
- **API_Wiki:** Clase que contiene los métodos y atributos necesarios para gestionar la comunicación con la API de Wikipedia.
- **Creación_rafaga_satisfactoria:** Clase que crea la interfaz necesaria para mostrar el mensaje de que se ha creado la ráfaga de imágenes correctamente.
- **Datos_imagen_wiki_love:** Clase que contiene los atributos necesarios para gestionar los datos recibidos de la API de Wiki Loves Monuments.
- **Datos_imagen:** Clase que crea la interfaz, métodos y atributos necesarios para introducir los datos oportunos para subir imágenes a Wikipedia.
- **Datos:** Clase que contiene atributos que se utilizan para mantener en memoria principal los datos necesarios para la lógica de la aplicación.
- **Descripcion_licencia:** Clase que gestiona la interfaz que muestra la descripción de cada una de las licencias existente.
- **Errores_api:** Clase que gestiona la interfaz que muestra los errores producidos por la comunicación con la API de Wikipedia.
- **Licencias:** Clase que gestiona la interfaz que muestras los distintos tipos de licencias que hay, además contienen los métodos y atributos necesarios para guardar dicha licencia.
- **Llamar_url:** Clase que gestiona la interfaz que muestra una página web en la que te puedes registrar para obtener una cuenta de Wikimedia Commons en caso de que no la tengas.
- **Login:** Clase que gestiona la interfaz, métodos y atributos necesarios para mostrar y recoger los datos necesarios para logear un usuario por primera vez.

Capítulo 4 Análisis

- Mapa_monumentos: Clase que contienen métodos, atributos y gestiona la interfaz que muestra un mapa con los monumentos recogidos.
- Mensaje_campos_obligatorios: Clase que gestiona la lógica e interfaz que muestra un mensaje con los campos obligatorios para continuar con la aplicación.
- Mensaje_informacion: Clase que gestiona la interfaz que muestra los distintos tipos de informaciones acerca de la aplicación.
- Menú_información: Clase que gestiona la interfaz que muestra el menú con los distintos tipos de información de interés para el usuario.
- Menú_love_monuments: Clase que gestiona los métodos, atributos e interfaz para la selección de las distintas opciones sobre los monumentos de Wiki Loves Monuments.
- Método_subida: Clase que gestiona los métodos, atributos e interfaz para subir una o varias imágenes a Wikimedia Commons.
- Modo_captura: Clase que gestiona la interfaz que muestra los distintos modos de manera para capturar imágenes.
- Obtener_coordenadas: Clase que gestiona la interfaz que muestra el GPS para obtener las coordenadas del monumento o imagen a subir.
- Selección_distancia_monumentos: Clase que gestiona la interfaz me muestra las distancias a modo de radio en la cual se buscaran los monumentos sin imágenes de los listados de Bienes de Interés Cultural BIC.
- Selección_imagen_individual: Clase que gestiona la lógica de la aplicación cuando se selecciona subir una imagen individualmente.
- Selección_imagenes_rafaga: Clase que gestiona la lógica de la aplicación cuando se quiere crear un conjunto de imágenes para su posterior subida.
- Selección_monumento: Clase que crea la interfaz, métodos y atributos para la selección de un monumento que no tiene imagen en los listados de Bienes de Interés Cultural BIC.

Capítulo 4 Análisis

- Selección_rafaga: Clase que crea la interfaz, métodos y atributos para la selección de una ráfaga de entre las existentes para su subida.
- Subir: Clase que contiene la lógica e interfaz para subir una imagen o ráfaga de imágenes a Wikimedia Commons.
- Tomar_foto: Clase que controla la interfaz que gestiona la cámara del teléfono.

A continuación se incluye el diagrama de clases conceptual que contiene dichas clases.

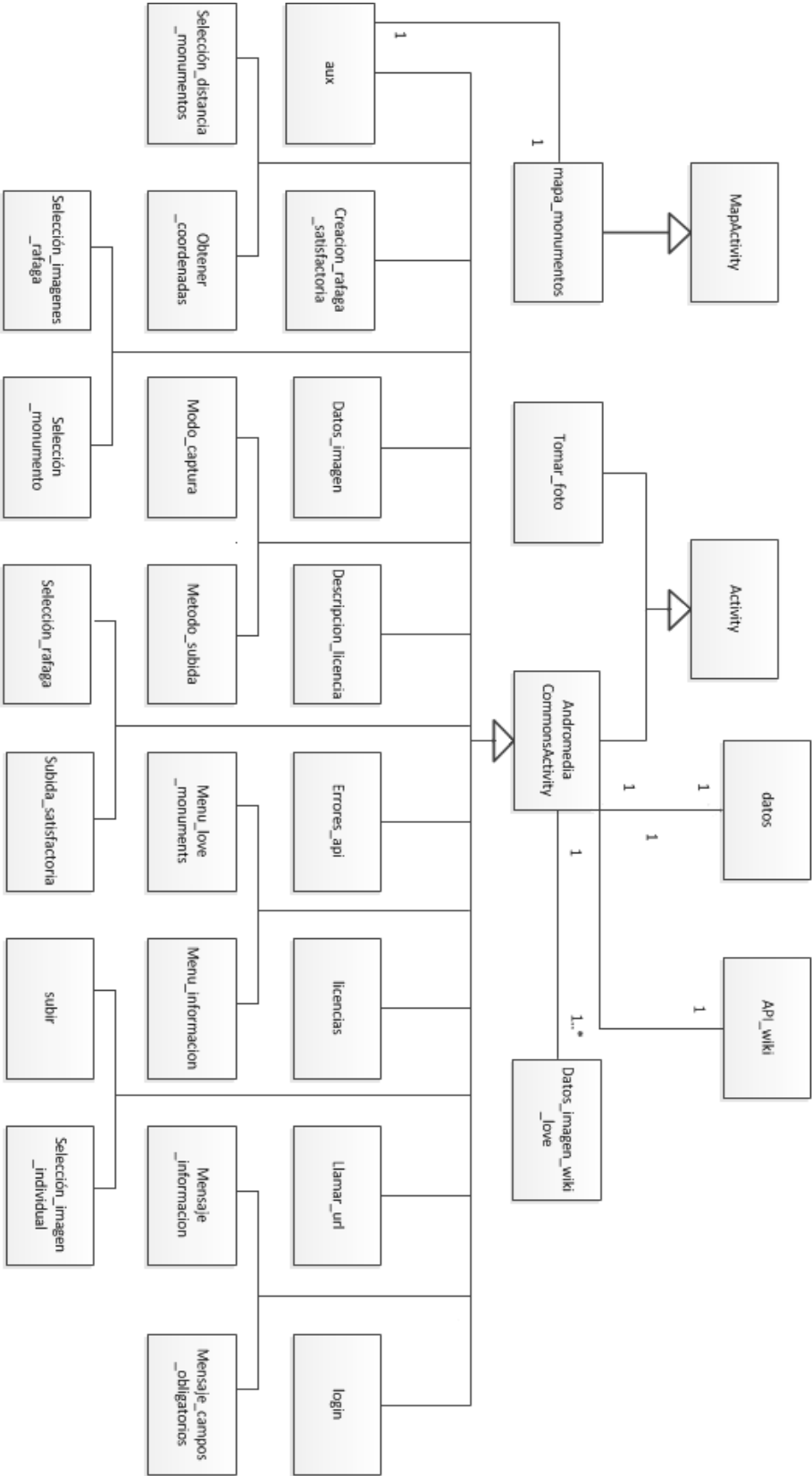


Figura 4.4 Diagrama Conceptual de Datos

4.4 Modelo de comportamiento

A continuación se incluye el modelo de comportamiento que especifica cómo debe de actuar el sistema, el cual consta de dos partes:

Diagrama de comportamiento y contrato de operaciones

Del mismo modo que en la sección anterior, he dividido el modelo de comportamiento en tres apartados, por lo tanto mostraré los diagramas de secuencias de: menú principal, interfaces y ver información.

4.4.1 Menú principal

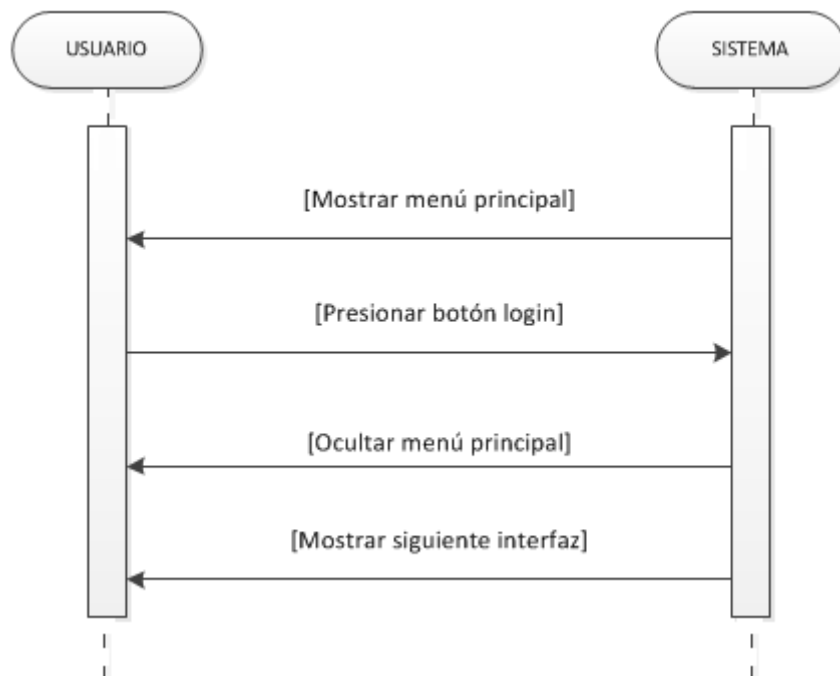


Figura 4.5 Diagrama de secuencia de sistema:Menu Principal

Contrato de las operaciones

Operación: Mostrar menú principal.

Responsabilidad: muestra el menú principal con las opciones “Login e “Información”.

Precondiciones: ninguna.

Postcondiciones: se ha cargado la interfaz para el menú principal.

Operación: Presionar botón “Login”.

Responsabilidad: Presionar botón “Login” y aparece un toolbar mientras se logea

Precondiciones: se ha mostrado el menú principal.

Postcondiciones: se ha presionado el botón login y ha aparecido un toolbar.

Operación: Ocultar menú principal.

Responsabilidad: oculta el menú principal para mostrar siguiente interfaz.

Precondiciones: se ha presionado el botón Login.

Postcondiciones: se han cargado los elementos necesarios para mostrar la siguiente interfaz.

Operación: Mostrar siguiente interfaz.

Responsabilidad: aparece la siguiente interfaz.

Precondiciones: ninguna.

Postcondiciones: se ha creado y mostrado la siguiente interfaz que corresponde.

4.4.2 Ver info

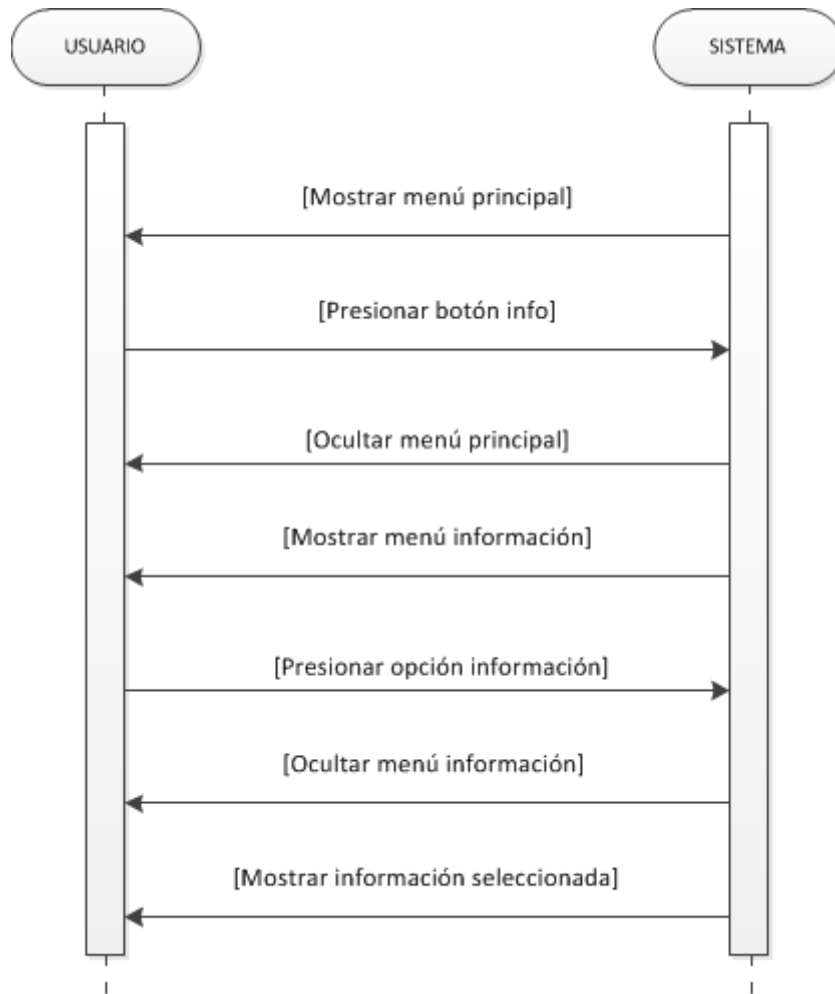


Figura 4.6 Diagrama de secuencia de sistema:ver información

Contrato de las operaciones

Operación: Ver información.

Responsabilidad: muestra el menú principal con las opciones “Login e “Información”.

Precondiciones: ninguna.

Postcondiciones: se ha cargado la interfaz para el menú principal.

Operación: Presionar botón “info”.

Responsabilidad: Presionar botón “info”

Precondiciones: se ha mostrado el menú principal.

Postcondiciones: se ha presionado el botón “info”.

Operación: Ocultar menú principal.

Responsabilidad: oculta el menú principal para mostrar la interfaz de opciones.

Precondiciones: se ha presionado el botón “info”.

Postcondiciones: se han cargado los elementos necesarios para mostrar la interfaz de opciones de información.

Operación: Mostrar menú de información.

Responsabilidad: seleccionar alguna de las opciones de información.

Precondiciones: ninguna.

Postcondiciones: se ha creado y mostrado la siguiente interfaz que corresponde.

Operación: Presionar opción información

Responsabilidad: Presionar sobre alguna de las opciones del menú de información

Precondiciones: se ha mostrado el menú de opciones de información

Postcondiciones: se ha presionado alguna de las opciones de información.

Operación: Ocultar menú información.

Responsabilidad: oculta el menú de información para mostrar la interfaz con la información seleccionada.

Precondiciones: se ha presionado una opción del menú de información”.

Postcondiciones: se han cargado los elementos necesarios para mostrar la interfaz con la información seleccionada.

Operación: Mostrar información seleccionada.

Responsabilidad: aparece la siguiente interfaz.

Precondiciones: ninguna.

Postcondiciones: se ha creado y mostrado la interfaz de información correspondiente.

4.4.3 Interfaces

Existen tres tipos de interfaces que se comportan de distinta manera y que trataremos de forma distinta, estas son:

- Las que esperan que se escoja una opción.
- Las que esperan que se rellene un formulario.
- Las que recuperan datos para el usuario

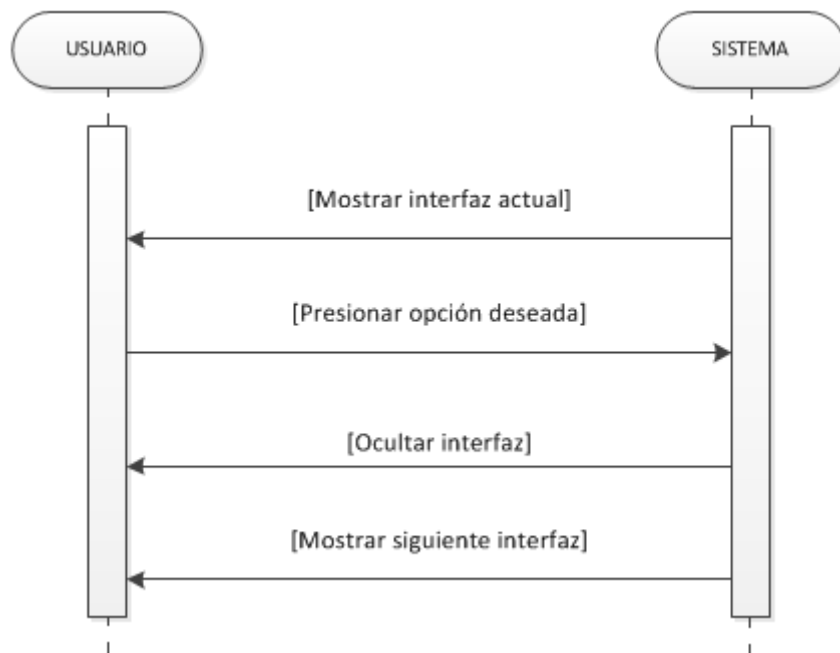


Figura 4.7 Diagrama de secuencia de sistema: escoger opción

Contrato de las operaciones

Operación: Mostrar interfaz actual.

Responsabilidad: muestra la interfaz actual con las opciones disponibles.

Precondiciones: ninguna.

Postcondiciones: se ha cargado la interfaz para la interfaz actual.

Operación: Presionar opción deseada.

Responsabilidad: Presionar opción deseada.

Precondiciones: se ha mostrado la interfaz actual.

Postcondiciones: se ha seleccionado alguna de las opciones disponibles.

Operación: Ocultar interfaz actual.

Responsabilidad: oculta la interfaz actual para mostrar siguiente interfaz.

Precondiciones: se ha seleccionado una opción determinada.

Postcondiciones: se han cargado los elementos necesarios para mostrar la siguiente interfaz.

Operación: Mostrar siguiente interfaz.

Responsabilidad: aparece la siguiente interfaz.

Precondiciones: ninguna.

Postcondiciones: se ha creado y mostrado la siguiente interfaz que corresponde.

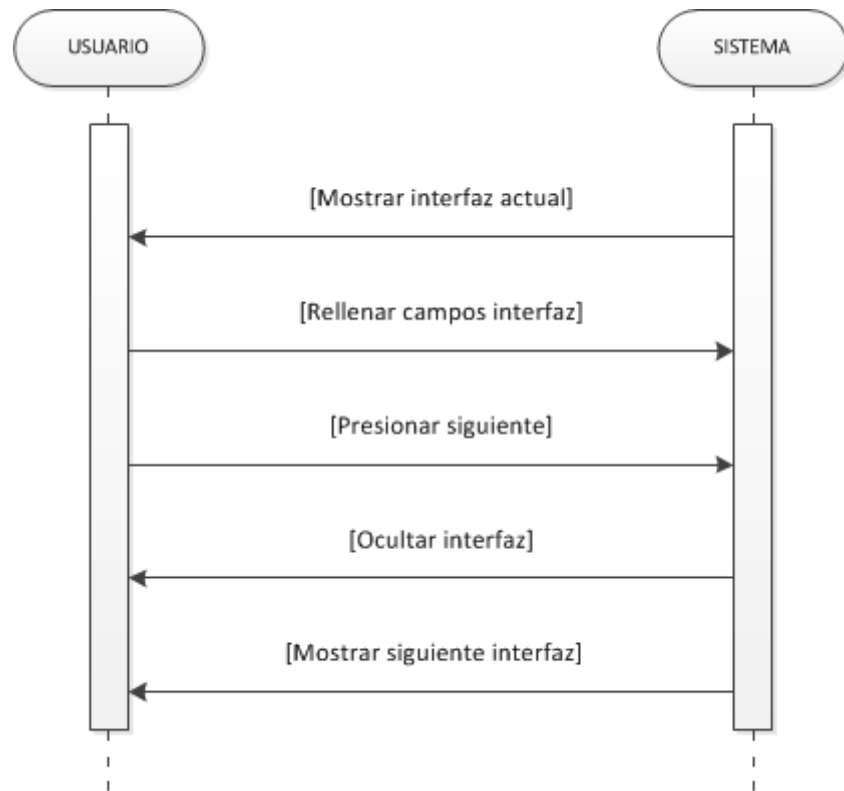


Figura 4.8 *Diagrama de secuencia de sistema:rellenar formulario*

Contrato de las operaciones

Operación: Mostrar interfaz actual.

Responsabilidad: muestra la interfaz actual con las opciones disponibles.

Precondiciones: ninguna.

Postcondiciones: se ha cargado la interfaz para la interfaz actual.

Operación: Presionar opción deseada.

Responsabilidad: Presionar opción deseada.

Precondiciones: se ha mostrado la interfaz actual.

Postcondiciones: se ha seleccionado alguna de las opciones disponibles.

Operación: Ocultar interfaz actual.

Responsabilidad: oculta la interfaz actual para mostrar siguiente interfaz.

Precondiciones: se ha seleccionado una opción determinada.

Postcondiciones: se han cargado los elementos necesarios para mostrar la siguiente interfaz.

Operación: Mostrar siguiente interfaz.

Responsabilidad: aparece la siguiente interfaz.

Precondiciones: ninguna.

Postcondiciones: se ha creado y mostrado la siguiente interfaz que corresponde

Contrato de las operaciones

Operación: Mostrar interfaz actual.

Responsabilidad: muestra la interfaz actual con las opciones disponibles.

Precondiciones: ninguna.

Postcondiciones: se ha cargado la interfaz para la interfaz actual.

Operación: Rellonar campos interfaz.

Responsabilidad: Rellonar campos interfaz.

Precondiciones: se ha mostrado la interfaz actual.

Postcondiciones: se han cumplimentados los campos de textos necesarios para pasar a la siguiente interfaz.

Operación: Presionar siguiente.

Responsabilidad: Presionar botón siguiente.

Capítulo 4 Análisis

Precondiciones: se han cumplimentados los campos de textos necesarios para pasar a la siguiente interfaz.

Postcondiciones: se han cargado los elementos necesarios para mostrar la siguiente interfaz.

Operación: Ocultar interfaz actual.

Responsabilidad: oculta la interfaz actual para mostrar siguiente interfaz.

Precondiciones: se ha presionado el botón siguiente

Postcondiciones: se han cargado los elementos necesarios para mostrar la siguiente interfaz.

Operación: Mostrar siguiente interfaz.

Responsabilidad: aparece la siguiente interfaz.

Precondiciones: ninguna.

Postcondiciones: se ha creado y mostrado la siguiente interfaz que corresponde.

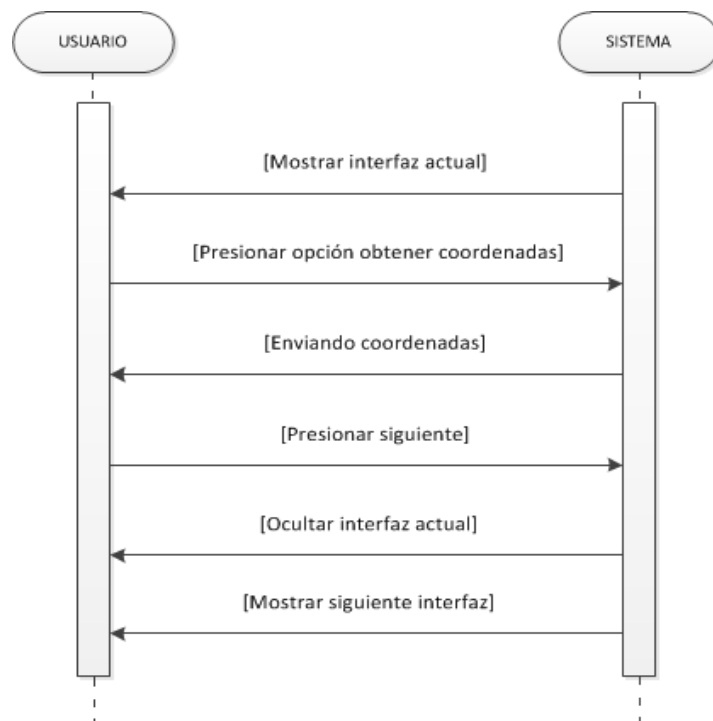


Figura 4.9 Diagrama de secuencia de sistema: obtener datos

Contrato de las operaciones

Operación: Mostrar interfaz actual.

Responsabilidad: muestra la interfaz actual con las opciones disponibles.

Precondiciones: ninguna.

Postcondiciones: se ha cargado la interfaz para la interfaz actual.

Operación: Presionar opción obtener coordenadas.

Responsabilidad: Presionar opción obtener coordenadas.

Precondiciones: se ha mostrado la interfaz actual.

Postcondiciones: se ha seleccionado presionado el botón obtener coordenadas.

Operación: Enviar coordenadas.

Responsabilidad: Enviar coordenadas.

Precondiciones: se ha pulsado el botón obtener coordenadas.

Postcondiciones: El sistema devuelve las coordenadas de la posición actual.

Operación: Presionar siguiente.

Responsabilidad: Presionar botón siguiente.

Precondiciones: se han obtenido las coordenadas de la posición actual.

Postcondiciones: se han cargado los elementos necesarios para mostrar la siguiente interfaz.

Operación: Ocultar interfaz actual.

Responsabilidad: oculta la interfaz actual para mostrar siguiente interfaz.

Precondiciones: se ha presionado el botón siguiente

Postcondiciones: se han cargado los elementos necesarios para mostrar la siguiente interfaz.

Operación: Mostrar siguiente interfaz.

Responsabilidad: aparece la siguiente interfaz.

Precondiciones: ninguna.

Postcondiciones: se ha creado y mostrado la siguiente interfaz que corresponde.

Capítulo 5

5 Diseño

En este capítulo hablo de las clases implementadas para llevar a cabo la aplicación.

Se elabora una pequeña descripción de que hace cada clase, los datos que contiene, los parámetros que utiliza, los métodos desarrollados y como interactúa con otras clases.

5.1 Diseño de la aplicación

Una vez especificado qué hace el sistema procedemos a detallar el diseño del sistema.

5.1.1 Descripción de las clases

5.1.1.1 Clase: *AndromediaCommonsActivity*

A small, faint diagram representing the class structure of AndromediaCommonsActivity. It appears to be a UML class diagram with a single class box and some internal details that are too small to read clearly.

Figura 5.1 *AndromediaCommonsActivity*

La clase *AndromediaCommonsActivity* es la clase principal de la aplicación, es la encargada de crear la interfaz principal y mantener todas las estructuras que se necesitan durante la ejecución de la aplicación.

Atributos:

public static API_Wiki usuario: Instancia de la clase API_WIKI que contiene los datos del usuario para logarse y subir imágenes.

public static datos: Instancia de la clase datos que contiene la información necesaria para almacenar durante la ejecución los datos de la o las imágenes a subir.

public static ArrayList<datos_imagen_wiki_love> lista_imagenes: Array de datos_imagen_wiki_love que contiene los datos de todos los monumentos obtenidos de la API.

public static ArrayList<String> datos_imagenes_rafaga: Array de string que contiene los path de todas las imágenes que se subirán de manera automática.

public static String texto_informativo="";

public static datos_imagen_wiki_love datos_monumento: Instancia de la clase datos_imagen_wiki_love que contienen los datos del monumento que vamos a subir la imagen.

Button btlogin: Instancia de la clase botón que se utiliza en la interfaz para logarse.

private ProgressDialog progressBarlogin: Instancia de la clase ProgressDioalog que muestra una animación gráfica mientras se realiza acciones en segundo plano.

private int progressBarStatus: Entero que permite determinar hasta cuando se mostrará el cuadro de dialogo.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Private void inicializar(): Método que establece todos los datos necesarios para mostrar la primera interfaz y logarse.

Capítulo 5 Diseño

Private int Click_Acceder()throws Exception: Evalúa si existen las credenciales, si existe, las lee y te logea automáticamente. En caso contrario, crea un intent que es el encargado de iniciar la nueva actividad.

Public void Click_informacion(View botton): Lanza el intent para llamar a la clase que crea la siguiente interfaz que tiene que mostrar.

Public String[] Leer_credenciales()throws Exception: Lee las credenciales del archivo donde están almacenadas para hacer login.

Private Boolean Existen_Credenciales()throws Exception: Evalúa si existen credenciales para logarse automáticamente.

5.1.1.2 Clase: Creación de ráfaga satisfactoria

...

Figura 5.2 Creación de ráfaga satisfactoria

La clase creación_rafaga_satisfactoria es la clase encargada de crear ejecutar la interfaz que se muestra cuando se sube de manera correcta una ráfaga de imágenes y además implementa el botón necesario para controlar el retroceso al menú principal.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Menu_Principal(View bottom): Método que recoge el evento click sobre el botón para llamar a la clase que ejecute el menú que se precisa.

5.1.1.3 Clase: *Datos imagen wiki love*

87

Figura 5.3 *Datos imagen wiki love*

La clase `datos_imagen_wiki_love` es la encargada de mantener en memoria principal los datos necesarios para poder subir al repositorio de monumentos la imagen deseada junto a todos los datos necesarios.

Atributos:

`public String municipio:` String que almacena el municipio del monumento recogido de la API.

`public String BIC:` String que almacena el tipo de monumento correspondiente recogido de la API. BIC->Bien Interés Cultural

`public String lista:` String que almacena el nombre del listado al que está asociado dicho monumento en la API.

`public String nom_imagen:` String que almacena el nombre de la imagen que está asociada al monumento en el repositorio donde está almacenado.

`public String lat:` String que almacena la latitud del monumento.

`public String lon:` String que almacena la longitud del monumento.

Capítulo 5 Diseño

public String distancia: String que almacena la distancia del monumento con respecto al punto en el que se tomaron las coordenadas con el dispositivo.

public String nom_monumento: String que almacena el nombre del monumento.

public String dominio: String que almacena el domino web donde está almacenado dicho monumento, su uso será necesario para la búsqueda de cualquier monumento, así como para la subida de imágenes asociadas.

5.1.1.4 Clase: Datos imagen

5

Figura 5.4 Datos imagen

La clase datos_imagen en la encarga de gestionar la interfaz y recoger los datos necesarios para poder subir la imagen requerida al repositorio de Wikimedia Commons deseado.

Atributos:

Private Intent intent: Los Intents son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Capítulo 5 Diseño

Métodos:

`Public void onCreate(Bundle savedInstanceState):` Método que nos permitirá realizar el arranque inicial de la interfaz.

`Private void rellenar_datos():` Método que recoge los datos introducidos en la interfaz y los almacena para su posterior subida.

`Public boolean campos_obligatorios:` Método que evalúa si los datos introducidos en los campos de la interfaz están de forma correcta y si falta alguno que debe ser necesario. Devuelve un booleano que indica si todo es correcto o no.

`Public void Click_Gps_No(View bottom):` Método que recoge el evento click de botón y que crea el intent para mostrar la siguiente interfaz.

`Public void Click_Gps_Si(View bottom):` Método que recoge el evento click de botón y que crea el intent para mostrar la siguiente interfaz.

5.1.1.5 Clase: *Datos*

El diagrama de la clase Datos, que debería estar aquí, no es visible en la imagen.

Figura 5.5 *Datos*

Clase que almacena en memoria principal los datos necesarios para mantener el funcionamiento coherente de la aplicación. Estos datos están asociados tanto con el usuario que se logea en la aplicación, como con la imagen o imágenes preparadas para ser subidas.

Capítulo 5 Diseño

Atributos:

String[] credenciales: Recoge las credenciales para logarse de los campos de la interfaz o del archivo donde están almacenadas en caso de que no sea la primera vez que el usuario accede.

String path: String que almacena la ruta de la imagen en el dispositivo para su posterior subida.

String nom_imagen: String que almacena el nombre que el dispositivo le da a la imagen que ha tomado.

String descripción: String que almacena la descripción de la imagen que se ha introducido en la interfaz.

String categoría: String que almacena la categoría de la imagen que se ha introducido en la interfaz.

String latitude: String que almacena la latitud de la imagen que se ha obtenido en la interfaz.

String longitude: String que almacena la longitud de la imagen que se ha obtenido en la interfaz.

String altitude: String que almacena la altitud de la imagen que se ha obtenido en la interfaz.

String descripcion_corta: String que almacena la descripción corta de la imagen que se ha introducido en la interfaz.

String fecha: String que almacena la fecha de la imagen que se ha introducido en la interfaz.

String licencia: String que almacena la licencia de la imagen que se ha introducido en la interfaz.

String path_rafaga: String que almacena el path de la ráfaga de imágenes que se ha seleccionado para ser subidas.

Capítulo 5 Diseño

String nombre_rafaga: String que almacena el nombre de la ráfaga de imágenes que se ha seleccionado para ser subidas.

Boolean subida_manual: Booleano que indica que la imagen va a ser subida de forma manual o automática.

Boolean ráfaga: Booleano que indica si se va a subir una imagen única o una ráfaga.

Boolean licencia_1: Booleano que indica que la licencia escogida es la 1.

Boolean licencia_2: Booleano que indica que la licencia escogida es la 2.

Boolean licencia_3: Booleano que indica que la licencia escogida es la 3.

ArrayList<String> nombre_lista_monumentos;

Boolean wikimonuments: Booleano que indica que la opción escogida es la de wikimonuments.

Boolean mapa: Booleano que indica que la opción escogida es la de mostrar mapa de monumentos.

String distancia: String que recoge la distancia que se va a tomar como radio para buscar los monumentos que se encuentran dentro de este rango.

Métodos:

Public datos(): Constructor de la clase datos. Inicializa todos los atributos.

Void ReiniciarDatos(): Método que restablece el valor inicial de los atributos.

5.1.1.6 Clase: Descripción licencia

Figura 5.6 Descripción licencia

Capítulo 5 Diseño

La clase descripción_licencia controla la interfaz que muestra la descripción de la licencia seleccionada.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Atras(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.7 Clase: Errores API

—

Figura 5.7 Errores API

La clase errores_API controla la interfaz que muestra la los distintos errores que pueden suceder al comunicarse con la API.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Atras(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.8 Clase: *Licencias*

371

Figura 5.8 *Licencias*

La clase licencias crea y controla la interfaz que permite al usuario escoger la licencia que se le otorgará a la imagen para ser subida.

Atributos:

Private Intent intent: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Siguiente(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

Private String Obtener_Datos_Rafaga(): Método que devuelve un string con los nombres de todas las imágenes de la ráfaga seleccionada.

Capítulo 5 Diseño

Private void Asignar_Licencia_Eleccion(): Método que asocia la licencia seleccionada a la imagen o ráfaga de imágenes asociada.

Public void Click_Check1(View view): Método que recoge el evento click sobre el check para indicar que tipo de licencia se ha escogido.

Public void Click_Check2(View view): Método que recoge el evento click sobre el check para indicar que tipo de licencia se ha escogido.

Public void Click_Check3(View view): Método que recoge el evento click sobre el check para indicar que tipo de licencia se ha escogido.

Public void Click_Licencia_1(View bottom): Método que crea el intent necesario para mostrar la interfaz seleccionada.

Public void Click_Licencia_2(View bottom): Método que crea el intent necesario para mostrar la interfaz seleccionada.

Public void Click_Licencia_3(View bottom): Método que crea el intent necesario para mostrar la interfaz seleccionada.

5.1.1.9 Clase: Llamar URL

=

Figura 5.9 Llamar URL

La clase llamar_url es la encargada de crear y controlar la interfaz que se muestra cuando el usuario no está registrado en wikimediacommons y quiere hacerlos.

Capítulo 5 Diseño

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

5.1.1.10 Clase: Llamar URL



Figura 5.10 Llamar url

5.1.1.11 Clase: Login

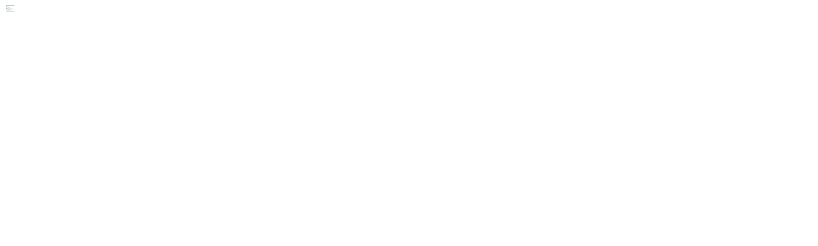


Figura 5.11 Login

La clase login crea y controla la interfaz con la que el usuario se logea.

Capítulo 5 Diseño

Atributos:

Private Bottom btAccederLogin: Instancia del botón que se crea en tiempo de ejecución.

Private int porgressBarStatus: Entero que permite determinar hasta cuando se mostrará el cuadro de dialogo.

Private ProgressDialog progressBarcredenciales: Instancia de la clase ProgressDioalog que muestra una animación gráfica mientras se realiza acciones en segundo plano.

Métodos:

Private int Click_Login()throws Exception(): Método que logea al usuario en el dominio correspondiente de Wikipedia.

Private void Escribir_Fichero(String credenciales)throws Exception(): Método que guarda en un fichero de forma permanente las credenciales del usuario para que no tenga que volver a escribirlas.

Private String Recoger_Credenciales(): Método que obtiene las credenciales para logarse y las devuelve en un string.

Public void Click_URL(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

Public void inicializar(): Método que permite mostrar el toolbar mientras se realiza el login en Segundo plano.

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

5.1.1.12 Clase: *Mapa monumentos*

87

Figura 5.12 *Mapa monumentos*

La clase `mapa_monumento` crea controla la interfaz que muestra el mapa de la ubicación donde te encuentras con los diferentes monumentos que tienes a tu alrededor.

Atributos:

`private Vector<String> info_toast:` Vector de string que contendrá el texto que se mostrara cuando se pulse sobre un monumento.

`private MapView mapView:` Instancia de la clase `MapView` que crea un mapa.

`private MapController myMapController:` Instancia de la clase `MapController` que ayuda a declarar parámetros como el zoom.

`private List<Overlay> mOverlays:` Lista de capas en las que mostrar los distintos monumentos en el mapa.

`private MyLocationOverlay mOverlayLocation:` Capa con la posición mí las coordenadas actuales del dispositivo.

`private aux x:` Instancia de la clase `aux` que hace de nexo de unión entre las clases `mapa_monumentos` y `WikimediacommonsActivity`.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Protected boolean isRouteDisplayed(): Devuelve un booleano que indica si el mapa lo que muestra son direcciones. Es un método redefinido de la clase MapActivity.

5.1.1.13 Clase: Mensaje campos obligatorios

=

Figura 5.13 Mensaje campos obligatorios

La clase mensaje_campos_obligatorios crea controla la interfaz que muestra el mensaje de que tipo de campo te has dejado sin rellenar en el formulario de información de imágenes.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Atras(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.14 Clase: Mensaje información

—

Figura 5.14 Mensaje información

La clase mensaje_informacion crea controla la interfaz que muestra los distintos mensajes de información que sirven para orientar al usuario sobre la aplicación.

Atributos:

Private Intent intent: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Atras(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

Public void Click_Principal(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.15 Clase: Menú información



Figura 5.15 Menú información

La clase mene_informacion crea controla la interfaz que muestra los distintos tipos de mensajes de información que sirven para orientar al usuario sobre la aplicación.

Atributos:

final Intent intent: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

ListView lv: Atributo que muestra un listado de monumentos en el cual se puede clicar cada uno de ellos.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

5.1.1.16 Clase: *Menú love monuments*

=

Figura 5.16 *Menú love monuments*

La clase `menú_love_monuments` crea y muestra la interfaz que permite elegir entre las dos opciones que tiene este módulo de la aplicación, seleccionar monumentos de un listado o ver monumentos sin imágenes en el mapa.

Atributos:

final `Intent intent`: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Métodos:

`Public void onCreate(Bundle savedInstanceState)`: Método que nos permitirá realizar el arranque inicial de la interfaz.

`Public void Click_Descargar_Monumentos(View bottom)`: Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

`Public void Click_Cargar_mapa(View bottom)`: Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.17 Clase: Método subida

«

Figura 5.17 Método subida

La clase método_subida crea y controla la interfaz que muestra los distintos modos de subir una imagen que existen:

Manual: introduciendo todos los datos uno a uno.

Semiautomático: Establece unos parámetros por defecto que el usuario a la hora de subir una imagen obviara y por lo tanto este método será más rápido que el anterior.

Atributos:

Private Intent intent: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Semi(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

Public void Click_Click_Manual(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.18 Clase: *Modo captura*

=

Figura 5.18 *Modo captura*

La clase modo_captura crea y controla la interfaz que muestra las distintas opciones de a la hora de capturar las imágenes, existen tres opciones:

Individual: Sube la imagen que desees añadiendo todos los datos necesarios, solo una imagen.

Ráfaga: Crea un archivo con un listado de imágenes y lo almacena a la espera de que el usuario lo suba.

Cargar ráfaga: Recoge el archivo de imágenes solicitado de entre los ofrecidos y sube todas las imágenes que contiene de forma automática.

Atributos:

Private Intent intent: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Public void Click_Manual (View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

Public void Click_Manual(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

Public void Click_Rafaga (View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.19 Clase: Obtener coordenadas

El diagrama de la clase 'Obtener coordenadas' no es visible en la imagen, pero se indica su presencia por un pequeño icono.

Figura 5.19 Obtener coordenadas

La clase obtener_coordenadas crea y controla la interfaz que obtiene las coordenadas de la posición actual del dispositivo y se la asigna a la imagen que se desea subir.

Atributos:

private LocationManager locMgr: Instancia de la clase LocationManager que proporciona APIs para determinar la ubicación y el soporte del dispositivo.

private LocationListener onLocationChange: Instancia de la clase LocationListener que se utiliza para recibir notificaciones del LocationManager cuando la ubicación ha cambiado.

Capítulo 5 Diseño

private TextView lon: TextView que muestra por la interfaz la coordenada de la longitud.

Private TextView lat: TextView que muestra por la interfaz la coordenada de la latitud.

private TextView alt: TextView que muestra por la interfaz la coordenada de la altura.

private Button bttriangulacion: Instancia de la clase boton para solicitar el modo de obtener las coordenadas y que se lanza en tiempo de ejecución;

private Button btgps: Instancia de la clase boton que se lanza en tiempo de ejecución e indica que las coordenadas se obtendrán mediante el GPS;

private Button bt_sig: Instancia de la clase boton que indica que ya se puede pasar a la siguiente pantalla y se lanza en tiempo de ejecución.

public Android.location.Location location: Variable para almacenar la posición actual obtenida del GPS o la triangulación;

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Private void inicializar(): Método que establece todos los datos necesarios para mostrar la primera interfaz y logarse.

Public void Click_Siguiente(View bottom): Método que recoge el evento click del botón y lanza el intent para mostrar la interfaz seleccionada.

5.1.1.20 Clase: Selección distancia monumentos

☐

Figura 5.20 Selección distancia monumentos

La clase selección_distancia_monumentos crea y controla la interfaz que muestra un listado con las distintas posiciones en las que tomando esta como radio, puedes encontrar monumentos sin imágenes.

Atributos:

Intent intent1: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Intent intent2: Los **Intents** son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Private int progressBarStatus: Entero que permite determinar hasta cuando se mostrará el cuadro de dialogo.

Private ProgressDialog progressBarMonuments: Instancia de la clase ProgressDialog que muestra una animación gráfica mientras se realiza acciones en segundo plano.

Capítulo 5 Diseño

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Private void inicializar(): Método que establece todos los datos necesarios para mostrar la primera interfaz y seleccionar los monumentos en un radio determinado.

Private int Obtener_Lista_Monumetros()throws Exception(): Método que recoge de la API de Wiki Loves Monuments los monumentos en un radio determinado a partir de la posición actual del dispositivo, y los almacena en una lista. Este método devuelve un entero que será usado para parar el progressbar que se ejecuta en primer plano mientras finaliza esta acción.

5.1.1.21 Clase: Selección imagen individual

—

Figura 5.21 Selección imagen individual

La clase selección_imagen_individual crea y controla la interfaz que muestra la galería de imágenes para que el usuario seleccione una.

Capítulo 5 Diseño

Atributos:

Intent intent: Los Intents son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Private int SELECT_IMAGE: Entero que indica que la interfaz que hay que mostrar es la de selección de imagen

Private Uri selectedImage:ruta de donde esta almacenada la imagen.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Private String Nombre_Imagen(String path): Método que se le pasa la ruta de la imagen y devuelve su nombre.

private void Elegir_Foto(): Método que lanza la interfaz para seleccionar la imagen de la galería del teléfono.

public String getPath(Uri uri): Método que recibe la uri de la imagen y la transforma en un path.

public void Click_ElegirOtra(View boton): Método que recoge el evento click del botón y muestra la interfaz que tienen asociada en su lógica.

public void Click_Siguiente(View bottom): Método que recoge el evento click del botón y muestra la interfaz que tienen asociada en su lógica.

public void onActivityResult(int requestCode, int resultCode, Intent data): Método que se lanza cuando una actividad lanzo otra y espera de esta una respuesta.

5.1.1.22 Clase: Selección imágenes ráfaga

82

Figura 5.22 Selección imágenes ráfaga

La clase selección_imagenes_rafaga crea y controla la interfaz que muestra la galería de imágenes para que el usuario seleccione las que desee de una en una.

Atributos:

Intent intent: Los Intents son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Private int SELECT_IMAGE: Entero que indica que la interfaz que hay que mostrar es la de selección de imagen

Private Uri selectedImage:ruta de donde esta almacenada la imagen.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Capítulo 5 Diseño

`Private String Nombre_Imagen(String path):` Método que se le pasa la ruta de la imagen y devuelve su nombre.

`private void Elegir_Foto():` Método que lanza la interfaz para seleccionar la imagen de la galería del teléfono.

`public String getPath(Uri uri):` Método que recibe la uri de la imagen y la transforma en un path.

`public void Click_ElegirOtra(View boton):` Método que recoge el evento click del botón y muestra la interfaz que tienen asociada en su lógica.

`public void Click_FinRafaga(View boton):` Método que recoge el evento click del botón y muestra la interfaz que tienen asociada en su lógica.

`public void onActivityResult(int requestCode, int resultCode, Intent data):` Método que se lanza cuando una actividad lanza otra y espera de esta una respuesta.

5.1.1.23 Clase: Selección monumento

◀

Figura 5.23 Selección monumento

La clase `selección_monumento` crea y controla la interfaz que muestra un listado de monumentos en un radio de n metros con respecto a la posición actual del dispositivo. Cada uno de estos elementos del listado es seleccionable.

Capítulo 5 Diseño

Atributos:

Final Intent intent:

ListView lv: ListView que contiene el nombre y distancia respecto a la posición actual de todos los monumentos que se muestra por pantalla.

String nom_monumento: String que contiene el nombre del monumento que se selecciona.

Métodos:

Los Intents son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Private void inicializar(): Método que establece todos los datos necesarios para mostrar la interfaz y seleccionar los monumentos

Private String Truncar_Decimales(String Distancia): Este método recibe un string y devuelve otro string con solo dos dígitos después de la coma.

5.1.1.24 Clase: Selección ráfaga

—

Figura 5.24 Selección ráfaga

Capítulo 5 Diseño

La clase selección _rafaga crea y controla la interfaz que muestra el listado de archivos de imágenes actuales disponibles para ser subidos mediante ráfaga.

Atributos:

Private Intent intent: Los Intents son conocidos como el mecanismo de paso de mensajes entre una Actividad a otra que puede trabajar entre nuestras propias Actividades o entre diferentes Aplicaciones.

ListView lv: Atributo que muestra un listado de ráfagas en el cual se puede clicar cada uno de ellos.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

5.1.1.25 Clase: Subida satisfactoria

...

Figura 5.25 Subida satisfactoria

La clase subida_satisfactoria crea y controla la interfaz que muestre un mensaje como que la o las imágenes se han subido correctamente, reinicia los datos necesarios para volver a realizar otra subida y te permite volver al menú principal.

Capítulo 5 Diseño

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

public void Click_Otra(View bottom): Método que recoge el evento click del botón y muestra la interfaz que tienen asociada en su lógica.

Private void Reiniciar_Datos(): Método que lanza reinicia el valor de todos los atributos de la clase datos para iniciar de nuevo la aplicación.

5.1.1.26 Clase: Subir

El diagrama de la clase Subir, que no es visible en la imagen, debería representar la estructura de la clase, incluyendo sus atributos, métodos y relaciones con otras clases.

Figura 5.26 Subir

La clase subir crea y controla la interfaz que sube las imágenes al repositorio correspondiente.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

5.1.1.27 Clase: *Tomar foto*



Figura 5.27 *Tomar foto*

La clase tomar_foto crea y controla la interfaz que permite al usuario tomar una imagen de un monumento dato y poder subir al repositorio correspondiente.

Métodos:

Public void onCreate(Bundle savedInstanceState): Método que nos permitirá realizar el arranque inicial de la interfaz.

Capítulo 6

6 Implementación

En este capítulo se habla de las características más relevantes a la hora de desarrollar la aplicación, centrándonos en los métodos más característicos y la API utilizada para recoger los datos de Wikipedia. Así como de la elección del lenguaje más apropiado para llevar a cabo la aplicación.

Para la implementación de la aplicación no me decidí por java por nada en particular, lo podría haber hecho por C, pero la mayoría de las aplicaciones están programadas en JAVA y además te ofrecen el entorno de desarrollo SDK para eclipse que te facilita mucho el trabajo, por lo que una vez tengas el código listo, solo tienes que compilar y automáticamente se te genera un archivo APK o archivo de instalación de Android, el cual solo tendrás que ejecutar en el móvil y ya tendrás la aplicación funcionando.

Las aplicaciones para Android se desarrollan en dos fases o partes, la interfaz que es un archivo .XML y la lógica que será un archivo .JAVA

6.1 API WIKIPEDIA

Solo fue empezar el proyecto y ya me encontré con uno de los dos grandes obstáculos que he tenido en todo este tiempo, ¿Cómo hago para comunicarme con Wikipedia para subir imágenes? Pues bien, buscando información y preguntando a diestro y siniestro entendí que existían las API'S, en este caso de Wikipedia.

Gracias a la API de Wikipedia, tenía la manera de poder logarme de manera automática por código. Ahora solo tenía que estructurar mi consulta con los datos pertinentes de forma correcta para pasárselos a la función deseada. Esta devuelve una serie de valores que son los que confirman que el usuario se ha logado con éxito y te permite poder realizar otra serie de funciones o actividades.

Tras muchos días investigando y haciendo pruebas descubrí que alguien había creado una clase JAVA con una serie de métodos y atributos que hacían lo que yo deseaba, comunicarme con la API. Desde ese momento, cogí dicha clase y la sometí a pruebas

Capítulo 6 Implementación

para ver si me resultaba útil. Pude comprobar que sí, y por lo tanto solo me quedaba adaptarla a mis necesidades.

6.2 Multilenguaje

La función multilenguaje para la aplicación una propiedad que la hace global y con una posibilidad mayor de expansión, ya que el número de usuarios que puede hacerse con ella será mucho mayor.

Actualmente solo recoge dos idiomas Castellano e Inglés, pero puede ampliarse sin ningún problema.

Para tener disponible distintos idiomas, solo hay que añadirlos a la carpeta de recursos asignada, traducir los textos necesarios y codificarla con el mismo nombre que tienen en el estándar de idiomas es lenguaje en cuestión.

Para que la aplicación aparezca en un idioma en concreto solo tendrás que cambiar el idioma del dispositivo desde su configuración.

6.3 Obtener coordenadas

La funcionalidad de obtener las coordenadas de la localización de un monumento o el punto exacto desde donde se ha tomado una foto es una de las cualidades más atractivas de esta aplicación, ya que ahorra al usuario todo el trabajo que tenía que realizar para poner las coordenadas en el método de subida tradicional.

Las tres funciones básicas de esta funcionalidad son la obtención de coordenadas mediante GPS, obtención de coordenadas mediante triangulación mediante red móvil y tratamiento de los datos para su posterior subida junto a la imagen y su salida por la interfaz.

6.4 Leer el contenido de la API y tratar los datos.

Esta funcionalidad es la más importante para el tratamiento de los datos de los monumentos, ya que recoge los datos solicitados de la API en un string, los trata y los devuelve estructurados de manera que su lectura sea lo más cómoda posible y además

Capítulo 6 Implementación

ayuda a agilizar el proceso ya que no hay que estar continuamente buscando entre un montón de datos sin conexión lógica aparente.

El otro método implementado en esta funcionalidad recoge los datos que devuelve la API, los separa por monumento y los almacena en un vector de monumentos.

Además recorre cada posición del vector de monumentos, leyendo sus datos tratándolos y asignándoselos a una instancia de la clase `datos_imagen_wiki_love` y esta a su vez metiéndola en una lista de clases de este mismo tipo.

Esto facilita muchas cosas, ya que a la hora de mostrar un listado de monumentos solo tenemos que recorrer cada nodo y seleccionar el atributo que se desee de la clase que está en dicho nodo.

6.5 Mapa con monumentos

La última funcionalidad a destacar es la que permite mostrar un mapa con todos los monumentos en un radio de distancia determinada que no tienen imagen, el método más importante para llevar a cabo esto es el que recibe los datos de un monumento concreto, crea un punto con sus coordenadas, crea una capa en la que metemos el nombre del monumento, el punto con las coordenadas y la distancia con respecto a la posición actual a la que se encuentra. Además añade un toast que mostrara el texto del monumento al que toques.

Además está el método que añade al mapa las distintas capas con los monumentos encontrados. Y una vez que tenemos todas las capas con todos los monumentos a mostrar, añadimos a la capa contenedora del mapa la posición nuestra que se refleja con un punto azul y todas las capas de monumentos que están contenidas en una instancia de la clase `Monumentos_Overlay`.

Capítulo 7

7 Pruebas

7.1 Incremento 1

- Problemas:

El principal problema que encontré en este incremento fue la comunicación con la API ya que carecía de cualquier formación o conocimiento de que como debía estructurar los datos y que tipo de consulta hacer para obtener resultado.

- Resolución a los problemas:

Este primer gran escollo lo resolví con un golpe de suerte y tras mucho investigar, ya que encontré implementada una clase que contenía los métodos necesarios para hacer la comunicación con la API.

- Pruebas:

Una vez concluido el primer incremento lleve a cabo las siguientes pruebas:

Comprobé que tras introducir la primera vez el usuario y la contraseña, el sistema almacena las credenciales en un archivo para no volver a solicitarlas.

También probé que si no es la primera vez que te logueas, no te vuelve a pedir las credenciales.

La selección de imágenes de la galería funcionaba correctamente ya que se habría la galería de imágenes y permitía escoger la que deseases. Una vez la tienes seleccionada la aplicación permite escoger el método de subida de la imagen manual que es introduciendo todos los datos y semiautomático que te permite obviar algunos datos para agilizar la subida.

También la elección de licencia para la imagen, ya que comprobé que mientras no seleccionas ninguna el sistema permite cambiar entre las licencias disponibles y ver su descripción si se requiere.

Por último comprobé que tras introducir los datos necesarios para la subida de la imagen a Wikimedia Commons y realizar este proceso, la imagen aparecía ya en el repositorio.

7.2 Incremento 2

- Problemas:

El principal problema de este incremento fue la dificultad que tenía para obtener las coordenadas en lugares parcialmente cerrados (con techo o paredes próximas).

- Resolución a los problemas:

Establecí otro proveedor de coordenadas además de GPS, este fue la conexión a internet. De esta manera cuando te encuentras en un sitio que no se obtiene señal GPS, se puede obtener las coordenadas por este otro medio.

- Pruebas:

En este paso las prueba realizadas en una primera instancia fueron crear una aplicación muy sencilla que tras presionar un botón, activara el GPS y devolviese las coordenadas de la posición actual del dispositivo.

Tras esto, me di cuenta de la problemática de si te encuentras en sitios cerrados, ya que en este caso existe la dificultad de encontrar señal GPS y por tanto añadí una pantalla más a la aplicación en la que te permitiera escoger entre triangulación mediante señal de internet y GPS. Comprobando que ambas funcionalidades devolvían las coordenadas de manera correcta.

Para finalizar el incremento, integre esta funcionalidad en las obtenidas en el incremento anterior y comprobé que tras la elección de la imagen, el sistema permitía obtener las coordenadas de la posición actual para vincularla a la imagen asociada. Además comprobé que los posibles errores que la aplicación podría dar en estos pasos estaban recogidos para informar al usuario en todo momento.

7.3 Incremento 3

- Problemas:

El problema que encontré en este incremento fue como almacenar todas las imágenes seleccionadas que quería subir en un momento determinado todas de golpe (es decir en una ráfaga).

Capítulo 7 Pruebas

- Resolución a los problemas:

Me cree un fichero de texto por cada ráfaga en el cual se introduce la ruta del directorio donde está ubicada la imagen. De esta manera cuando se va a subir la ráfaga, solo hay que leer del fichero la ruta e ir a buscar la imagen a su lugar correspondiente.

- Pruebas:

Con las pruebas realizadas pude comprobar que la información para las imágenes seleccionadas para la ráfaga son almacenadas en un fichero en el cual se van concatenando las rutas de las imágenes para su posterior recuperación.

También comprobé que el archivo que se crea tiene el nombre de la ráfaga con la fecha y hora de su creación y que una vez se selecciona uno del listado, lo sube de manera automática al tú repositorio de Wikimedia Commons.

7.4 Incremento 4

- Problemas:

En este incremento el principal problema que encontré fue como catalogar los monumentos y gestionar su subida al repositorio correspondiente ya que existe alguna disparidad entre ellos; algunos monumentos están alojados en anexos y otros en listas, cabe destacar que la lógica para el mantenimiento de ambos repositorio es distinta.

Otro problema que me surgió durante este desarrollo fue que cuando alojaba una imagen en su anexo o lista correspondiente, Wikipedia me decía que necesitaba permiso de administrador para realizar esta acción.

- Resolución a los problemas:

Desarrollé el código de la subida de imágenes de monumentos en función de si lo tenía que subir a un anexo o a una lista. Para averiguar dónde estaba contenido dicho monumento tuve que tratar los datos contenidos en la API y leer un campo en el cual ponía la url de su ubicación y es ahí donde indica si está en un anexo o una lista.

El problema del administrador de Wikipedia se solventó solo ya que nunca me volvió a parecer, y desde entonces no ha vuelto a dar problemas.

Capítulo 7 Pruebas

- Pruebas:

Para el desarrollo en este incremento, comprobé que una vez recogía los datos según las coordenadas de la posición actual del dispositivo, los monumentos verdaderamente están cerca de lo que indica la aplicación. Por otro lado, también comprobé que cuando un monumento pertenecía a un listado de bienes de interés cultural, los subía a dicho listado y en el monumento que corresponde. Además he comprobado que en posiciones simuladas con la máquina virtual que si encuentras monumentos que no estén en listado de bienes de interés cultural, si no en anexos, también sube la imagen y actualiza dicho anexo.

Para finalizar como en los anteriores incrementos, añadí esta nueva funcionalidad a la aplicación e hice una prueba completa para asegurarme de que todo funcionaba correctamente.

7.5 Incremento 5

- Problemas:

El principal problema que encontré fue a la hora de construir el mapa con cada uno de los monumentos y mi posición actual, ya que desconocía la forma en la cual se representan cada una de las distintas coordenadas que hace referencia a cada monumento en el mapa.

- Solución:

Buscando información encontré una web en la que explicaban como hacer algo parecido. Dicha web explicaba como mostrar tu posición actual y cualquier posición que quisieses, de esta manera solo tuve que extrapolar dicho ejemplo a mi caso y en vez de añadir una sola posición, realice un buche con todos los monumentos que me devuelve la consulta a la API.

- Pruebas: En este punto, con la aplicación muy avanzada, aunque no terminada, ya ofrecía la oportunidad de poder realizar un uso más o menos normal de la aplicación, por lo que gracias a Manuel Palomo Duarte que me consiguió un listado

de personas que son miembros de WikiMedia España, pudimos hacer que ellos fuesen los primeros en darnos su opinión y posibles correcciones a realizar de la aplicación.

Lo más significativo que se encontró durante estas pruebas es que durante el desarrollo no restringí la aplicación para que no subiera datos reales al repositorio de cada usuario. Esto provocaba que subieras imágenes que en realizada no querías tener ya que o no representaban lo que decían ser o simplemente eran imágenes tomadas de cualquier objeto o lugar que no tuviese relevancia alguna.

Tras realizar dichos usuarios las pruebas pertinentes durante varios días, volví a modificar el código para que realmente no se subiesen imágenes y se pudiese realizar el ciclo completo de cada acción sí que esto tuviese efecto alguno en el repositorio de cada usuario.

7.6 Prueba Completa

- Problemas:

En este último incremento no tuve problemas ya que lo único novedoso con respecto a los anteriores es que cuando subes una imagen a Wikimedia Commons, también creas una referencia a ella y la almacenas en una base de datos interna para tener un registro de las imágenes que has subido con la aplicación.

- Pruebas:

Como la finalidad de este último incremento es tener posicionadas en el mapa todas las imágenes que has subido a tu repositorio de Wikimedia Commons, las pruebas que realicé fueron comprobar que si una imagen la subía al repositorio de Wikimedia Commons sin especificar las coordenadas, pues no la insertaba en la BBDD, además, si insertabas una imagen y luego la borrabas de la galería entonces no te la muestra en el mapa ya que no existe el fichero al que hace referencia.

También comprobé que los datos que muestra cuando pulsas sobre un monumento son los que le corresponde.

Y finalmente una vez estuvieron todos los incrementos finalizados y con la aplicación funcional, he realizado una prueba de toda la aplicación en la que subí imágenes a mi

Capítulo 7 Pruebas

repositorio de Wikimedia Commons en modo individual, en modo ráfaga. Busque monumentos sin imágenes y los mostré en el mapa y subí una imagen de un monumento sin imagen sin que se encontrase ningún tipo de problema.

Capítulo 8

8 Conclusiones

En este capítulo describo el por qué decidí realizar una aplicación Android y las posibles ampliaciones que se le podrían hacer a mi aplicación.

8.1 Desarrollo del proyecto

Escogí una aplicación para Android porque mi principal objetivo era aprender a desarrollar aplicaciones para este tipo de plataforma y porque me parecía una idea atractiva para desarrollar como proyecto final de carrera.

A día de hoy puedo decir que he aprendido nociones básicas de JAVA, ya que para desarrollar aplicaciones tampoco es necesario un conocimiento muy extenso del lenguaje, además he aprendido a utilizar las librerías de JAVA para Android con algo de soltera, por lo menos las que he utilizado. También XML para desarrollar las interfaces de la aplicación, así como el manejo de base de dato SQLite para Android aunque finalmente no lo haya puesto en práctica.

Como conclusión puedo decir que me siento satisfecho de lo que he realizado y aunque me ha llevado mucho tiempo y dedicación, no se me ha hecho pesado en absoluto ya que es algo que encuentro curioso e intrigante.

En un futuro, me gustaría tener la oportunidad de poder desarrollar alguna aplicación a nivel profesional ya que considero que puede ser algo productivo.

8.2 Ampliaciones

Desde que he terminado el desarrollo he pensado en posibles mejoras y ampliaciones y aunque al principio cegado por la obsesión de ponerle un punto final mi mente no me permitía ver más allá de esto, a día de hoy he encontrado varios puntos con los que ampliar la aplicación. A continuación detallo alguno de ellos:

- En la pantalla de mostrar mapa de monumentos, en vez de mostrar un toast con el nombre del monumento y la distancia a la que se encuentra de la posición actual que ocupas cuando haces click sobre él, que te permita abrir la cámara y tomar una imagen de dicho monumento para subirla al listado o anexo correspondiente.

Capítulo 8 Conclusiones

El principal problema de esta opción es que puedes seleccionar un monumento que este a 1 km por ejemplo y que como es lógico dudo que se le pueda tomar una foto desde esa distancia y selecciones otra cosa a lo que le hagas la foto y la subas en su nombre. Aunque supongo que esto siempre es algo que debe de controlar el usuario. Además para eso están los moderadores y administrados de Wikipedia.

- Otra posible ampliación sería que una vez has subido una imagen de un monumento, en la pantalla de subida satisfactoria, además de mostrarte un botón que te lleva a la pantalla de elegir opción, te mostrase otro botón que te permitiese abrir un navegador web y te redirigiese al anexo o listado correspondiente donde se encuentra dicho monumento.
- Y la última que se me ocurrió y a modo de ofrecer todo lo que ofrece la aplicación de Wikimedia Commons, sería que te mostrase tanto en un listado como en un mapa todas las imágenes que tienes en tu repositorio. Independientemente del móvil o PC que utilices para subirlos al repositorio, es decir, que te descargue y muestre las imágenes que subiste en cualquier día.

8.3 Comparativa con Wikimedia Commons

El proceso de subida de imágenes al repositorio que existe actualmente en el web <https://commons.wikimedia.org/wiki/Special:UploadWizard>, necesita 4 clics para subir una imagen más los necesarios para rellenar los datos necesarios.

Mientras que con mi aplicación necesitarías 6 clics para subir una imagen más necesarios para rellenar los datos necesarios. Estos dos clics de más que debemos de realizar con mi aplicación no son un problema ni un ralentizado del proceso, ya que lo que estos dos clics ofrecen son una de las mayores ventajas con respecto a la web. Estamos hablando del posicionamiento mediante GPS o triangulación.

Mientras que la subida individual de imágenes podemos decir que es muy parecida con respecto al número de clics a realizar, la mayor ventaja de mi aplicación viene dada cuando queremos subir un número más elevado de imágenes. A continuación detallo el

Capítulo 8 Conclusiones

número de clics necesarios para subir 10 imágenes desde la web y 10 imágenes desde mi aplicación:

La subida desde la web requiere 4 clics por cada imagen. 1 Para empezar el proceso, 1 para pasar al siguiente paso cuando seleccionas la imagen, 1 para la selección de la licencia y 1 para dar la orden de subir la imagen. Esto multiplicado por 10 imágenes, nos da un total de 40 clics.

Mi aplicación necesita 1 clic para seleccionar la subida mediante ráfaga, 1 clic por cada imagen que seleccionas, 1 clic para finalizar el proceso de creación de ráfaga, 1 click para la selección de la licencia que se les otorga a las imágenes, 1 clic para volver al menú principal, 1 click para seleccionar la carga de ráfagas, 1 click para seleccionar la ráfaga deseada y 1 clic para iniciar el proceso de subida. Por lo tanto para mi aplicación subir 10 imágenes se necesitan 7 clic para completar el proceso más 10 de la elección de las 10 imágenes, lo que hace un total de 17 clics.

Comparando ambos métodos de subida con 10 imágenes obtengo una mejora del 57,5% en el número de clics necesarios para la subida de dichas imágenes. Y si se subiesen 100 imágenes obtendría una mejora de casi el 75% en el número de clics.

Este porcentaje se traduce en el tiempo que transcurre entre una subida y la siguiente ya que mientras en la web hay que realizar el proceso desde el principio por cada imagen, con la aplicación esto se agiliza, ya que las imágenes se seleccionan todas antes de la subida, y la subida se hace al final con el total de las imágenes.

Y para finalizar con comparativa con la subida por la web, mientras que las imágenes subidas por la web no permiten el posicionamiento geográfico de manera directa, mi aplicación posiciona las imágenes y te las almacena en una base de datos para que puedas visualizarlas en su un mapa personalizado con tus imágenes en el punto exacto donde las tomaste.

9 Apéndice

9.1 Apéndice A

9.1.1 Instalación

La instalación es tan sencilla como otra cualquiera para esta plataforma, solo tienes que descargarte de GooglePlay la aplicación en caso de que estuviese o de cualquier otra web, y una vez descargada solo pulsar instalar y una vez finalizado el proceso, ya se puede disfrutar de ella.

9.2 Apéndice B

9.2.1 *Guía de usuario*

El primer requisito para manejar la aplicación es logarse en caso de tener cuenta. Si no la tienes la aplicación te facilita una pantalla en la que puedes hacerlo.

Una vez te has logado puedes escoger entre varias opciones como son subir una foto individualmente, crear una ráfaga de imágenes, subir una ráfaga de imágenes en el caso de que ya se haya creado alguna o buscar monumentos sin imágenes en Wikipedia.

Para la primera opción, subir imagen individual, existen dos métodos de subida, manual y semiautomático que en lo único que difieren es que en el primero el usuario está obligado a introducir todos o la mayoría de los datos solicitados, mientras en el modo semiautomático la aplicación te pone muchos de ellos por defecto.

Para la segunda opción, crear una ráfaga, los pasos a seguir son muy parecidos a los de la primera opción con la salvedad de que puedes escoger varias imágenes para subirlas.

La tercera opción, subir ráfaga, te muestra un listado de las ráfagas disponibles para poder ser subidas, el método de subida es similar al semiautomático.

La cuarta opción muestra dos posibles alternativas, buscar monumentos sin imágenes en un radio de terminado de metros y subir una imagen asociada a él, o ver en un mapa la posición que ocupan estos monumentos.

Y por último existe la opción de ver las imágenes que has subido al repositorio de Wikimedia Commons con el dispositivo actual posicionadas en el mapa.

A continuación detallo como es el proceso estándar para el uso de la aplicación y las ilustraciones pertinentes para cada caso:

9.2.1.1 Subida de imagen individual

Una vez hemos instalado la aplicación la primera pantalla que nos aparece es la de logearnos, lo cual podemos hacer pulsando acceder.



Imagen guía 1

Si es la primera vez que accedemos a la aplicación tendremos que introducir las credenciales para utilizar la aplicación. En caso de no tener cuenta en Wikimedia Commons solo tendremos que pulsar el botón Registrarse en Wikimedia Commons y se nos abra el navegador en la página exacta para crear nuestra cuenta.



Imagen guía 2

Apéndice

Si lo deseamos antes de entrar en las distintas funciones que ofrece la aplicación podemos ver información básica sobre esta pulsando el botón azul de la pantalla inicial donde pone “info” y nos aparecera el siguiente menu, en el cual encontraras los requisitos de la aplicación, advertencias sobre su uso, mis datos e información sobre el concurso Wiki Loves Monuments.

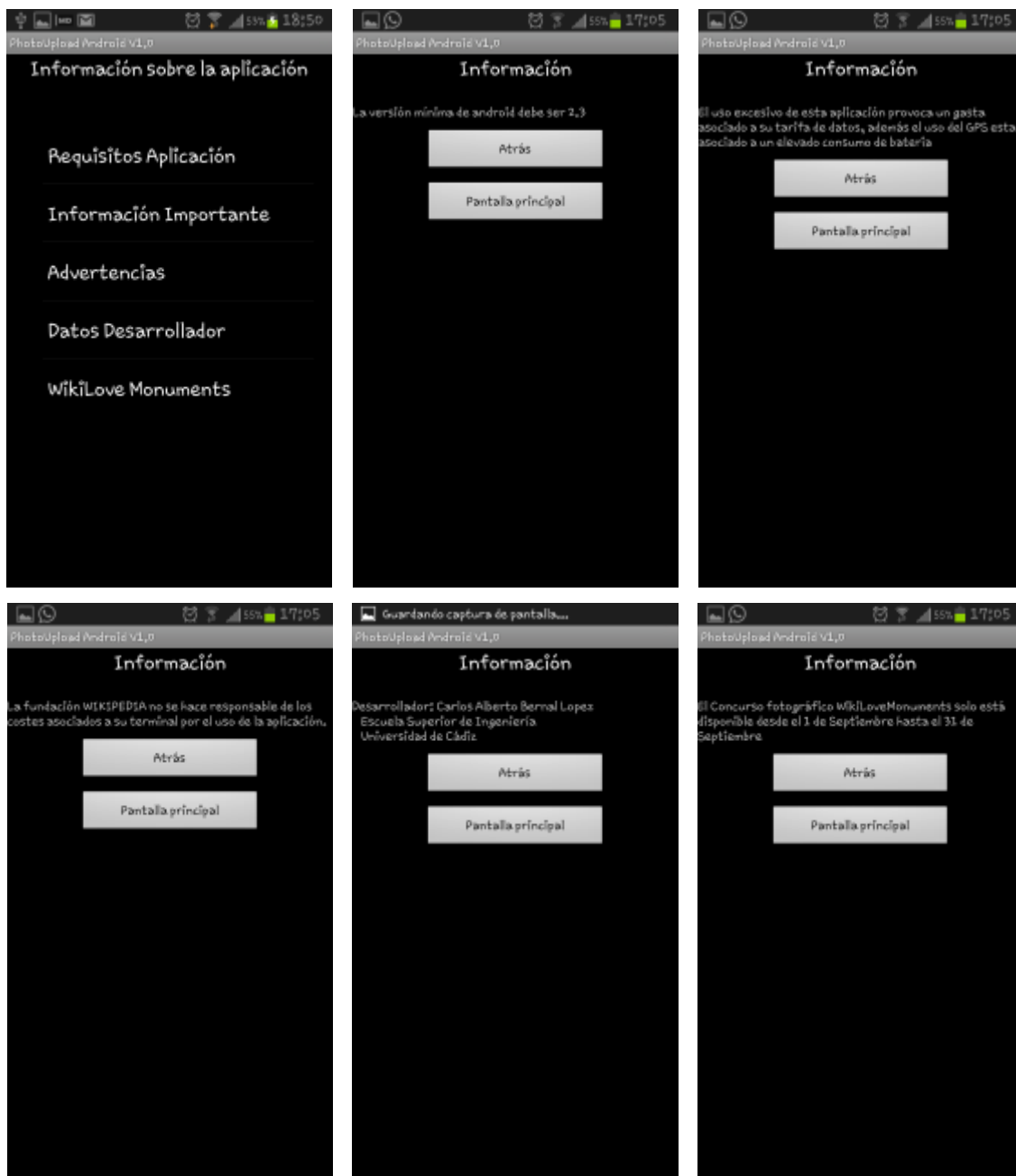


Imagen guía 3

Apéndice

Tras ver la información básica presionamos el botón de acceder de la pantalla inicial y nos encontramos con el menú de las distintas opciones que nos ofrece la aplicación.

El menu principal consta de elegir imágenes para subir a Wikimedia Commons de manera individual, por ráfagas (más de una imagen a la vez), buscar monumentos sin imágenes para actualizar el listado o anexo de bienes de interes cultural al que pertenece y por último ver las imágenes que ya hemos subido a Wikimedia Commons con nuestro dispositivo.

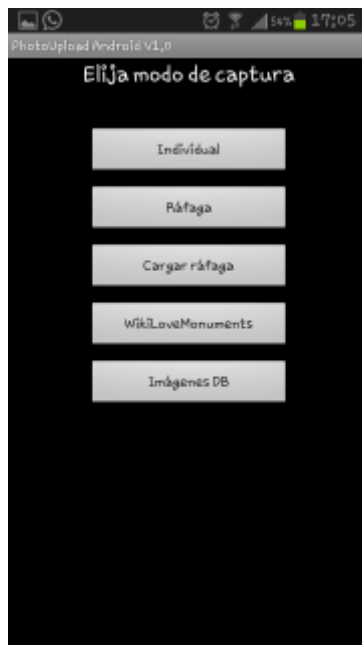


Imagen guía 4

Si seleccionamos la opción de subir una imagen a nuestro repositorio de Wiklimedia Commons de manera individual, se nos abrira la galería de imágenes y tras seleccionar una de ellas, la veremos en nuestra patalla junto a dos botones que nos permiten avanzar hasta el siguiente paso o por el contrario volver a seleccionar otra imagen.



Imagen guía 5

Una vez hemos seleccionado la imagen que deseamos subir a nuestro repositorio de Wikimedia Commons, se nos mostrará un menu con los distintos tipos de licencias que la aplicación ofrece y la descripción asociada a cada una de ellas.

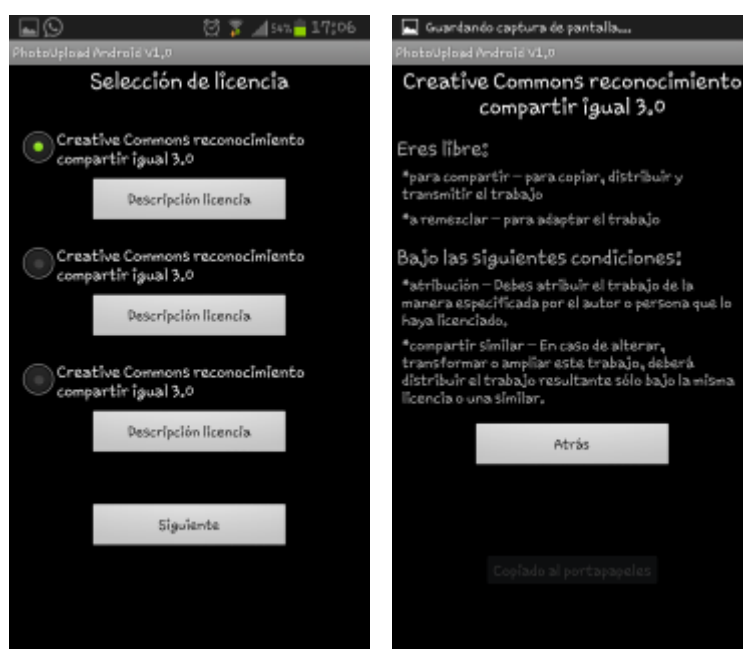


Imagen guía 6

Apéndice

Tras seleccionar la licencia que le queremos dar a la imagen, pasamos a rellenar los campos del formulario necesarios para la subida de la imagen.

Estos campos necesarios son el título, la descripción y la fecha que viene dada por la imagen en caso de que esta la tenga almacenada. Pero además le podemos dar una descripción más corta a modo que nos hagamos una rápida idea de que es la imagen, un listado de categorías donde encuadrarlas, y posicionarla mediante GPS o triangulación.

A screenshot of the 'PhotoUpload Android V1.0' app interface. It features a dark-themed form with white text and input fields. The fields are labeled 'Título', 'Descripción corta', 'Descripción', 'Categoría', and 'Fecha'. The 'Fecha' field is a date picker showing '24 Nov 2013'. Below the date picker is a question '¿Desea posicionar mediante GPS la imagen?' with 'Sí' and 'No' buttons. A 'Añadir Categoría' button is next to the 'Categoría' field.

Imagen guía 7

Tras introducir los datos, si pulsamos que deseamos posicionar la imagen, nos aparecerá una pantalla como esta, en la que tenemos que escoger entre triangulación o señal GPS.


A screenshot of the 'Obtener posición actual' screen in the app. It has a dark background with white text. At the top, it says 'Obtener posición actual'. Below that, a message reads: 'Si está en un lugar cerrado pulse triangulación mediante la red, de lo contrario pulse posicionamiento GPS'. There are two buttons: 'Triangulación' and 'GPS'. At the bottom, it shows 'Longitud: Buscando...', 'Latitud: Buscando...', and 'Altitud: Buscando...'.

Imagen guía 8

Apéndice

Lo ideal es que si estamos en un espacio abierto, escojamos la señal GPS, mientras que si nos encontramos en lugares en los que la señal GPS llegue con dificultad y tenemos una buena conexión a internet debemos escoger la triangulación.

Una vez la aplicación recupere las coordenadas nos aparecerá un botón para poder avanzar en el proceso. Ya que mientras obtenemos las coordenadas la aplicación queda parcialmente bloqueada puesto que no puedes ir ni hacia atrás ni hacia adelante.



Imagen guía 9

Una vez ya nos aparece el botón solo tenemos que pulsar el botón de subir imagen y esperar a que este proces finalice.

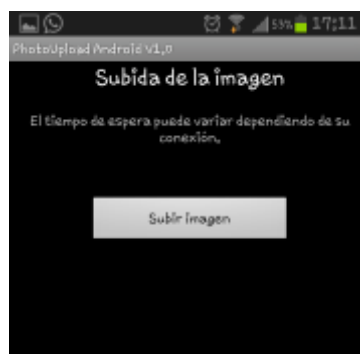


Imagen guía 10

Una vez finalice el proceso pulsa el botón Menú Principal y volveras al menú donde escoger las distintas opciones.



Imagen guía 11

9.2.1.2 Creación de una ráfaga

Si seleccionamos la opción de crear una ráfaga, nos aparecerá la galería de imágenes y una vez seleccionemos la que deseemos nos aparecerá esta como cuando elegimos una imagen en el modo individual, pero además tenemos la opción de incluir otra imagen más en la ráfaga, y esto sucederá con tantas imágenes como deseemos.

9.2.1.3 Selección de una ráfaga

Tras tener ráfagas de imágenes creadas, si pulsamos en la tercera opción del menú principal, se nos mostrará un listado con las ráfagas almacenadas disponibles para ser subidas a nuestro repositorio de Wikimedia Commons



Imagen guía 12

Apéndice

Tras seleccionar la ráfaga deseada, en este caso solo existe 1, tendremos que escoger la licencia y pulsar subir ráfaga a nuestro repositorio tal y como hicimos con la imagen individual.

9.2.1.4 Buscar Monumentos Wiki Loves Monuments

Si en el menú principal pulsamos la cuarta opción, Wiki Loves Monuments, se nos mostrará otro menú con las opciones de subir imágenes de monumentos que carezcan de ésta al listado o anexo correspondiente de bienes de interes cultural o ver los monumentos sin imágenes que existen en un radio cercano a nuestra posición.



Imagen guía 13

Apéndice

Si pulsamos en actualizar el listado de monumentos, tendremos que escoger una distancia en torno a nosotros en metro sobre la que hacer la búsqueda.



Imagen guía 14

Tras esto, nos aparecera un listado con los monumentos dentro del rádio seleccionado que no contienen imágenes.

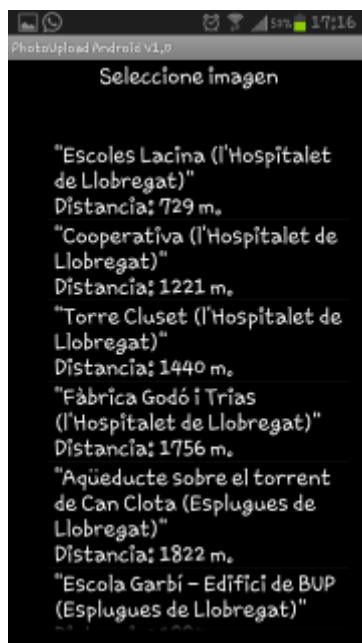


Imagen guía 15

Apéndice

Tras pulsar sobre uno de los monumentos del listado, se abre la cámara, podrás tomar la imagen que desees de el monumento seleccionado y realizar los mismo pasos que para la subida de imagen individual.

Sin embargo, si pulsas en el menú de Wiki Loves Monuments la opción de ver la ubicación de monumentos, tendremos que volver a seleccionar una distancia en torno a nuestra posición y tras esto, nos aparecerá un mapa con los distintos monumentos cercanos a nuestra posición. Y para tener más información, dependiendo de la versión del sistema operativo Android que utilices, podrás ver un mensaje emergente en la pantalla que te indica que monumento es el que has tocado y a que distancia de ti está.



Imagen guía 16

9.2.1.5 Imágenes DB

La última funcionalidad que nos presta la aplicación es la de ver las imágenes que hemos subido a nuestro repositorio de Wikimedia Commons con nuestro dispositivo.

Si pulsas el último botón en el menú principal, nos aparecerá un mapa con nuestra posición actual representada por un punto azul y los iconos que representan las imágenes que hemos tomado en un punto en el que las tomamos.

Apéndice

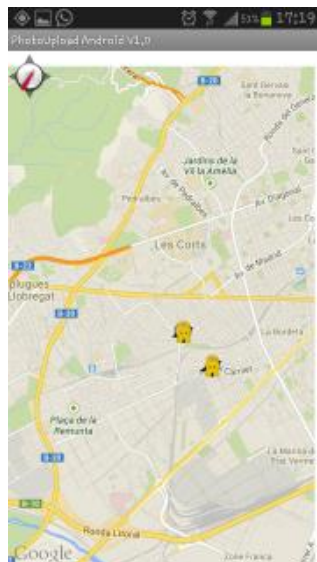


Imagen guía 17

Y si pulsas sobre uno de los iconos que representan las imágenes que hemos tomado, accederas a una nueva pantalla en la que se muestra la imagen que subiste, la descripción que le otorgaste y la fecha de la imagen.



Imagen guía 18

10 Bibliografía

[1] Página oficial de Wikipedia. Autor: Wikipedia

<http://es.wikipedia.org/wiki/Wikipedia>

En esta dirección habla sobre que es Wikipedia y sus datos más relevantes

[2] Página oficial de Wikimedia Commons. Autor: Wikipedia

http://es.wikipedia.org/wiki/Wikimedia_Commons

En esta dirección describe que es Wikimedia Commons y un poco de historia sobre sus orígenes.

[3] Página oficial de Wiki Loves Monuments. Autor: Wikipedia

<http://www.wikilm.es/>

En esta web encontrarás información sobre el concurso Wiki Loves Monuments.

[4] Curso de programación para android. Autor: Sgoliver.net blog.

http://www.sgoliver.net/blog/?page_id=3011

[5] Curso para Andorid. Todo lo que necesitas para empezar Autor: Maestro de la web

http://www.maestrosdelweb.com/editorial/curso-Android/?utm_source=mdw&utm_medium=home&utm_campaign=guias

[6] Curso de programación en Android. Autor: Slideshare.

<http://www.slideshare.net/techmi/curso-desarrollo-en-Android>

[7] Programación de interfaz Andorid. Autor: Androidsis

<http://www.Androidsis.com/programacion-Android-iii-interfaz/>

[8] Programación en Android desde cero. Autor: Jesusenlanet.

<http://jesusenlanet.blogspot.com.es/2012/01/tutorial-manual-howto-programacion.html>

Desde la cuarta dirección hasta la octava se encuentran guías, foros y explicaciones sobre la programación en android con distintos ejemplos y código para que pueda ser reutilizado

Bibliografía

[9] Código cliente para la API de Wikipedia. Autor: MediaWiki

http://www.mediawiki.org/wiki/API:Client_code

[110] Código API de Wikipedia en JAVA. Autor: wiki-java

<http://wiki-java.googlecode.com/svn/trunk/src/org/wikipedia/Wiki.java>

En esta dirección se encuentra implementada la API de Wikipedia que he utilizado para desarrollar las diferentes funcionalidades de la aplicación.

[11] Localización geográfica en Android. Autor: Sgoliver.

<http://www.sgoliver.net/blog/?p=1932>

Explicación de cómo usar el dispositivo GPS de nuestro teléfono con aclaraciones sobre el código que se utiliza para llevarlo a cabo.

[12] Ejemplo de barra de progreso. Autor: Mkyong.

<http://www.mkyong.com/Android/Android-progress-bar-example/>

En esta URL se encuentra una breve explicación de cómo implementar una barra de progreso para cuando la aplicación esté realizando alguna acción en segundo plano y no debamos tomar nada mientras.

[13] Uso de la memoria externa del dispositivo. Autor: Androideity.

<http://Androideity.com/2011/11/03/escribiendo-archivos-en-la-memoria-externa-sd-card/>

En esta URL se habla de cómo utilizar la memoria externa para almacenar información desde una aplicación que desarrollemos y pequeños ejemplos de almacenar información en ficheros tanto en memoria interna como externa.

[14] Uso de mapsvew en android. Autor: Androcode.

<http://androcode.es/2011/12/usando-mapviews-desde-el-principio/>

En esta dirección habla de cómo implementar un mapa con las distintas capas necesarias para introducir distintos elementos (coordenadas).

- Libros interesantes para el desarrollo de la aplicación:

Bibliografía

[15] Wikipedia de la A a la W de Tomás Saorín. ISBN: 978 84 9029 012 5

[16] ANDROID 4 PRINCIPIOS DEL DESARROLLO DE APLICACIONES JAVA,
ISBN: 9782746082861

- Urls de referencia para el uso de la aplicación

https://es.wikipedia.org/wiki/Categoría:Bienes_de_interés_cultural_de_España

En esta url se encuentran todas las subcategorías de bienes de interés cultural.

<https://commons.wikimedia.org/wiki/Portada?uselang=es>

Esta url es la página principal de Wikimedia Commons en la cual se encuentra tú repositorio de imágenes en caso de que tengas cuenta asociada.

11 Agradecimientos

Quiero agradecer a Manuel por la ayuda y conocimientos prestados a mi causa siempre que lo he necesitado, por la atención prestada durante este tiempo y por ofrecerme el desarrollo de una aplicación con la que he disfrutado desde el primer momento, la cual me ha enseñado un nuevo lenguaje de programación y ha ampliado mis conocimientos.

A los miembros de Wikimedia España que me han ayudado dando su punto de vista durante las pruebas de la aplicación.

Agradecer también a mi Nieves su esfuerzo por aguantar que no le hiciese compañía ni compartiera con ella muchísimas tardes por quedarme encerrado delante del ordenador y por las tardes soporíferas tardes de sofá que ha tenido mientras yo seguía encerrado.

A mis padres por siempre estar encima mío para que no me desanimara y recordarme que tenía que entregarlo o entregarlo.

Y por último a una persona muy especial que ya no está y sé que le hubiese gustado verlo.

12 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

[<http://fsf.org/>](http://fsf.org/)

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement

made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an

"aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

GNU Free Documentation License

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document

under the terms of the GNU Free Documentation License, Version 1.3

or any later version published by the Free Software Foundation;

with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU

Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the

Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.